



Large Scale Choreographies for the Future Internet

ICT IP Project

Deliverable D1.2

CHORReOS Perspective on the Future Internet and Initial Conceptual Model

<http://www.choreos.eu>

THALES



INSTITUT FÜR INFORMATIK
UND KOMPUTERWISSENSCHAFT
AN DER UNIVERSITÄT ZÜRICH



template v0



Project Number	:	FP7-257178
Project Title	:	CHOReOS – Large Scale Choreographies for the Future Internet

Deliverable Number	:	D1.2
Title of Deliverable	:	CHOReOS perspective on the Future Internet and initial conceptual model
Nature of Deliverable	:	Report
Dissemination level	:	Public
Version	:	VA.0
Contractual Delivery Date	:	31 March 2011
Actual Delivery Date	:	11 April 2011
Contributing WP	:	WP1
Editor(s)	:	Marco Autili, Davide Di Ruscio (UDA)
Author(s)	:	Marco Autili, Davide Di Ruscio, Amleto Di Salle (UDA); Nikolaos Georgantas, Sara Hachem, Valérie Issarny (INRIA); Athanasios Parathyras, Lefteris Trimintzios (VTRIP); Darius Silingas (BPI); James Lockerbie, Neil Maiden (CITY); Amira Ben Hamida (Petals), Antonia Bertolino, Guglielmo De Angelis (CNR-ISTI); Andrea Polini (Unicam); Dionysis Athanasopoulos, Apostolos Zarras (UOI).
Reviewer(s)	:	Hugues Vincent (THALES), Valerie Issarny (INRIA), Darius Silingas (NME).

Abstract

The D1.2 deliverable outlines the CHOReOS perspective on the Future Internet and its conceptualization. In particular, the deliverable focuses on:

- Definition of the Future Internet and related Future Internet of Services and (Smart) Things, as considered within CHOReOS, further stressing the many dimensions underpinning the Ultra-Large Scale of the Future Internet;
- Definition of the initial conceptual model of the CHOReOS Service-Oriented Architecture (SOA) for the Future Internet, identifying the impact of the ULS dimensions upon the traditional SOA paradigms and associated infrastructure.

Keyword list

Ultra-Large Scale systems, Future Internet, Future Internet of Services, Future Internet of Things, Choreography-centric Service Oriented Computing and Architecture, Choreography, Conceptual Model, Meta-modeling, Software Architecture, Requirements, Model-driven Engineering, BPMN2.

Document History

Version	Changes	Author(s)
V0	First version of outline, integration of material from the DoW, and notes from the Evaluation Summary Report, specific issues/questions addressed during the Evaluation Hearing, EC Questions and Recommendations, first feedback from the Collaboration meeting (19-20 October in Brussels)	- Marco Autili (UDA)
V0.1	Second version of outline embedding comments from Valérie and Hugues	- Marco Autili (UDA)
V1.0	Final Outline	- Marco Autili (UDA)
V1.1	First Draft including a preliminary version of the Introduction, Section 2 and Section 4	- Marco Autili, Davide Di Ruscio, Amleto Di Salle (UDA)
V1.2	Refined Draft including a preliminary version of Section 3	- Riccardo Mazza (WIND) - Gianmarco Panza, Teodoro De Giorgio, Maurilio Zuccalà (CEFRIEL)
V1.3	First integration of all contributions	- All Authors
V1.3.1	Revised/Integrated Chapter 2	- Valérie Issarny, Sara Hachem, Georgantas Nikolaos (INRIA) - Apostolos Zarras, Dionysis Athanasopoulos, Panos Vassiliadis (UOI)
V1.3.2	Overall revision/integration of all chapters but Chapter 2	- Marco Autili, Davide Di Ruscio, Amleto Di Salle (UDA)
V1.3.3	Overall revision/integration of all chapters	- Marco Autili, Davide Di Ruscio, Amleto Di Salle (UDA) - Valérie Issarny, Sara Hachem, Georgantas Nikolaos (INRIA)
V1.4	Chapter 4 has been completely rewritten and has been split into “Baseline for the CHOReOS Conceptual Model” (Chapter 4) and “Initial version of the CHOReOS Conceptual Model” (Chapter 5)	- Marco Autili, Davide Di Ruscio (UDA)
V1.4.1	Overall revision. Many contents have been added to Chapter 5 concerning the initial CHOReOS Conceptual Model. A new section has been added to discuss future steps towards the CHOReOS Architectural Style	- Marco Autili, Davide Di Ruscio (UDA)
V1.4.2	Integration of contributions into Chapter 4	- Marco Autili, Davide Di Ruscio (UDA)
V1.4.3	Overall revision of Chapter 4	- Marco Autili, Davide Di Ruscio (UDA) - Neil Maiden, James Lockerbie (CITY)

V1.4.4	Integration of contributions into Chapter 4 and 5 from Petals	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA) - Amira Ben Hamida (Petals) - Antonia Bertolino, Guglielmo De Angelis, Daniela Mulas (CNR-ISTI) - Andrea Polini (Unicam) - Apostolos Zarras (UOI)
V1.4.5	Addressing of comments from Valérie Issarny (SL)	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA)
V1.4.6	Integration of contents from Daniel Macêdo Batista and overall revision by UDA	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA) - Daniel Macêdo Batista (USP)
V1.4.7	Integration of contents from UOI, CNR and VTRIP, and overall revision by UDA.	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA) - Apostolos Zarras (UOI) - Antonia Bertolino, Guglielmo De Angelis, Daniela Mulas (CNR-ISTI) - Lefteris Trimintzios, Athanasios Parathyras (VTRIP)
V1.4.8	Removal of Chapter 3. Dynaroute scenario has been included in Chapter 2. Overall revision by UDA.	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA)
V1.4.9	Integration of new contribution by INRIA for Chapter 2 and Section 4.2	<ul style="list-style-type: none"> - Marco Autili, Davide Di Ruscio (UDA) - Valérie Issarny, Sara Hachem, Georgantas Nikolaos (INRIA)
V2.0	Overall revision of document	<ul style="list-style-type: none"> - Valérie Issarny (INRIA)
V2.2	Comments by V. Issarny addressed	<ul style="list-style-type: none"> - M. Autili, D. Di Ruscio (UDA)
V2.3	Integration of comments by A. Zarras, V.Issarny, N. Georgantas	<ul style="list-style-type: none"> - M. Autili, D. Di Ruscio (UDA)
V2.4	Integration of the revised Discovery Concern section by A.Zarras	<ul style="list-style-type: none"> - M.Autili, D.Di Ruscio (UDA)
V2.5	Integration of comments by N. Georgantas and A. Ben Hamida	<ul style="list-style-type: none"> - M.Autili, D.Di Ruscio (UDA)

Document Review

Review	Date	Ver.	Reviewers	Comments
Outline	14/10/10	1.0	All authors	OK by all
Draft 1	01/04/11	2.0	V. Issarny	Commented version by V.Issarny
Draft 2	04/04/11	2.2	M. Autili, D. Di Ruscio	Comments by V. Issarny addressed
QA	10/04/11	3.0	V. Issarny (SL), Marco Autili (WPL+DL), Hugues Vincent (PL), Darius Silingas (NME - External Reviewer)	Complete version released for final review
PTC	11/04/10	V3	PTC	/

Glossary, acronyms & abbreviations

Item	Description
BPEL	Business Process Execution Language
BPMN	Business Process Modeling Notation
DL	Deliverable Leader
DOW	Description of Work
FIA	Future Internet Assembly
MDE	Model Driven Engineering
NEXOF	NESSI Open Service Framework
NEXOF-RA	NESSI Open Service Framework-Reference Architecture
PL	Project Leader
SL	Scientific Leader
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture
SoaML	Service oriented architecture Modeling Language
SOC	Service-Oriented Computing
TDD	Test-Driven Development
UPDM	Unified Profile for DoDAF and MODAF
USDL	Unified Service Description Language
ULS	Ultra-Large Scale
WP	Work Package
WPL	Work Package Leader
WSCI	Web Service Choreography Initiative
WS-CDL	Web Service – Choreography Description Language
WSDL	Web Service Description Language
V&V	Validation & Verification

Table of Contents

1. Introduction	1
2. Future Internet: The CHOReOS Definition	3
2.1. The Many Facets of the Future Internet.....	3
2.1.1. Core Domains of the Future Internet.....	3
2.1.2. Cross-domain Challenges of the Future Internet	6
2.1.3. CHOReOS Focus on the Future Internet of Services and (Smart) Things	9
2.2. Evolution towards the Future Internet of Services and Smart Things	9
2.2.1. Population of Services & Smart Things.....	10
2.2.2. Complexity of Services & Smart Things.....	11
2.2.3. Functional Diversity of Services and Smart Things.....	11
2.2.4. Concluding Remarks.....	12
2.3. Future Internet Requirements from the DynaRoute Use Case	12
3. The NEXOF Reference Architecture	17
3.1. The NEXOF Reference Architecture in a Nutshell	17
3.1.1. Guidelines and Principles.....	19
3.1.2. Reference Specification	19
3.1.3. Reference Model.....	20
3.1.4. Reference Model vs Reference Specification	21
3.2. Overview of the Composition Concern.....	22
4. Initial Version of the CHOReOS Conceptual Model.....	24
4.1. Composition Concern	26
4.1.1. Structure View.....	27
4.1.2. Data Flow.....	27
4.2. Presentation Concern	31
4.3. Analysis Concern	31
4.3.1. Structure View.....	32
4.3.2. Data Flow.....	33
4.4. Discovery Concern.....	33
4.4.1. Structure View.....	34
4.4.2. Data Flow.....	35
4.5. Message Concern.....	39
4.5.1. Structure View.....	40
4.5.2. Data Flow.....	41
4.6. Service Concern	43
4.7. Resource Concern	44
4.8. Governance and Management Concern	44
4.8.1. Structure View.....	45
4.8.2. Data Flow.....	46
4.9. Security Concern	48
5. Specifying and Realizing Concepts of the Initial Conceptual Model	49
5.1. Baseline for Domain Expert Specification of Requirements.....	49
5.2. Baseline for Service Specifications	53
5.2.1. SoaML – Service-oriented architecture Modeling Language.....	53
5.2.2. USDL - Unified Service Description Language	54
5.3. Baseline for Choreography Specifications	55
5.4. Baseline for Choreography Execution.....	57
5.5. Baseline for Governance and V&V	58
6. Conclusions.....	60
7. References.....	61

1. Introduction

The **Future Internet**, as a particular case of **Ultra-Large Scale (ULS)** systems [ULS06], constitutes a futuristic vision of a yet-to-come Internet whose “scale changes everything”. As a matter of fact, the understanding of the Future Internet is receiving significant attention (e.g., [TFI10]) since, as for any new domain, there is not yet a well established characterization of it. In the context of the European Economic Recovery Plan, it is worth mentioning the Future Internet Assembly (FIA¹) that groups over 100 projects to coordinate their R&D activities to foster a strong European footprint on Future Internet. The main objective is developing open, standardized, cross-sector service *infrastructures*. From a European policy perspective, sectors such as healthcare, mobility, environment and energy management are prime candidates to benefit from novel “smart” – Internet-empowered – infrastructures, which will facilitate the rapid take-up and adoption of services by millions of users.

The CHOReOS project positions itself in the above vision of the Future Internet, whilst focusing on the Future Internet of Services and (Smart) Things. In this scenario, the highly-scalable choreographies that CHOReOS wants to investigate play a central role and, as such, their key characteristics and requirements need to be identified by fully articulating the significance and centrality of *collaboration of services* in the context of Future Internet systems.

As reported in CHOReOS deliverable [D1.1], a number of valuable conceptual and architectural characterizations have been proposed in the literature for modeling/describing service-oriented systems. Roughly speaking, it can be said that all these approaches strive towards the same goal even though at different levels of abstraction and with different specific purposes. The most recent approach is the NEXOF Reference Architecture (NEXOF-RA)². NEXOF-RA defines a *pattern-based reference architecture* for *SOA infrastructures* within the Future Internet and, as part of it, defines a *conceptual model* specifying the first-level entities that constitute these infrastructures, the facilities they provide, as well as relationships among them [SLC10]. However, NEXOF-RA does not emphasize the concept of choreography, which is abstractly considered as a specialization of service composition, without providing a clear definition of it. Instead, CHOReOS aims at investigating the impact of the Future Internet’s ULS on service-oriented systems realized as choreographies of services.

Within this line of research, the purpose of this deliverable is to present the CHOReOS perspective on the Future Internet and related challenges, together with introducing the resulting conceptual model for the choreography-based Future Internet of Services, with a special focus on the supporting infrastructure. Accordingly, the deliverable is structured as follows:

- Chapter 2 provides a characterization of the Future Internet, and related Future Internet of Services and Things, as considered within CHOReOS, further stressing the many challenges that this poses for the development of systems.
- Chapter 3 then outlines the reference architecture and the embedded conceptual model defined by the NEXOF-RA project, which serve as a baseline to the definition of the CHOReOS Conceptual Model for the choreography-based Future Internet of Services.
- Chapter 4 defines the initial conceptual model for CHOReOS, building upon the Future Internet characterization given in Chapter 2, and the baseline reference architecture surveyed in Chapter 3. The conceptual model provides a high level common ground by capturing the relevant entities/concepts, and relationships among them, underlying the choreography-based Future Internet, and will serve as input to the development of the RTD work packages WP2-3-4. As usual, the conceptual model is defined following an iterative process that starts with the representation of preliminary identified concepts, which is the focus of this deliverable. From that perspective, in its initial version, the CHOReOS conceptual model provides some extensions to NEXOF-RA. In particular, it emphasizes service choreography that is becomes a first-class concept in CHOReOS. As the project progresses, CHOReOS concepts will be refined/extended according to the outcomes of RTD work packages WP2 to WP5.

¹ <http://www.future-internet.eu/>

² <http://www.nexof-ra.eu/>

- Chapter 5 further describes baselines for different languages, models, technologies that can be used to specify and realize concepts embedded within the CHOReOS initial conceptual model.
- Finally, Chapter 6 concludes the deliverable, with a summary of its contribution and follow-up work to be undertaken next, in particular based on the initial conceptual model definition.

2. Future Internet: The CHOReOS Definition

Over the years, Internet has become the most important networking infrastructure, providing an integrated entity enabling: creating, contributing, sharing, using, and integrating information and knowledge by all. As a result, the Internet is changing at a fast pace and is called to evolve into the *Future Internet*, i.e., *service- and self-aware federated networks* that provide built-in and integrated capabilities such as: contextualization, reliability, robustness, mobility, security, service support, and self-management of communication resources and services [TFI10].

As mentioned in the introduction, the Future Internet Assembly (FIA) aims at fostering the development of open, standardized, cross-sector service platforms so that many sectors such as healthcare, mobility, environment and energy management can benefit from novel and smart infrastructures supporting service exploitation by millions of users. Still, several research challenges must be overcome to attain such a goal due to the shortcomings of today's Internet, especially regarding the lack of built-in facilities to support non-basic functionalities required by a service-aware Internet [FIA09]. The work described in [ETP09] further illustrates the overall objectives and ambitions underlying the trend towards the Future Internet in Europe. In particular, the authors of [ETP09] propose a strategy and action plan that will make the Future Internet an industrial, economic and societal success for Europe. Potential problems and risks associated with the Future Internet as well as potential opportunities/benefits of a new networking approach are discussed in [FIAMANA09]. Some concrete usage scenarios of Future Internet are also proposed in [FIAFCN07, FISO09, FIRE09, FISE09, RWI09]. For each scenario, the authors give description, functional requirements, potential barriers and problems, R&D challenges, as well as potentials for business innovation.

Beyond initiatives supported by the EC (see <http://www.future-internet.eu>), the Future Internet has become the main focus of several research and development initiatives all over the world, including initiatives in the USA (e.g., the NSF Future Internet Design (FIND) initiative - <http://www.nets-find.net>), China (e.g., China Next Generation Internet (CNGI) - <http://www.cstnet.net.cn/english/cngi/cngi.htm>; and Ministry of Science and Technology 863 program - http://www.most.gov.cn/eng/programmes1/200610/t20061009_36225.htm), Korea (e.g., <http://fif.kr>) and Japan (e.g., <http://akari-project.nict.go.jp/eng/overview.htm>). This effectively leads to a noticeable number of projects whose aim is to provide and implement a vision leading to the establishment of a *global-scale, interoperable, and secure smart Internet*.

However, despite and maybe because of the tremendous interest for the Future Internet, the term is very generic and no common definition has been adopted yet. Still, the Future Internet may be defined in terms of the core constituents of today's Internet and foreseen evolution, as outlined in Section 2.1. This further leads us to highlight core challenges and requirements posed upon the computing and networking environments, which in particular have to be faced by CHOReOS. Then, Section 2.2 more specifically concentrates on an early characterization of the Internet of Services and Smart Things that is focused upon within CHOReOS and that is further illustrated in Section 2.3 using one of the application scenarios that are targeted by CHOReOS, i.e., the Dynaroute use case.

2.1. The Many Facets of the Future Internet

Based on the broad and mostly informal definition of the Future Internet available in the literature, this section provides an overview of the many facets that must be considered in the realization of the Future Internet. Our presentation is independent of the more specific focus of CHOReOS so as to put our RTD work into the perspective of the overall Future Internet vision.

2.1.1. Core Domains of the Future Internet

Based on the Future Internet literature and in particular the one originating from the aforementioned international initiatives, the Future Internet, resulting from the evolution of today's Internet, can be defined as the union and cooperation of the *Internet of Content*, *Internet of Services* and *Internet of Things*, supported by an expanding network infrastructure foundation. Those core domains, elements of which we find already in today's Internet, are not fully established yet and will emerge with the foreseen evolution of services, content, objects and networks as discussed in what follows.

Internet of Content

Today, we can notice the following trends regarding the Internet of content, which are emerging to reshape the current Internet content and are paving the way towards the future. The first trend is *social networking*, supported by the use of Web 2.0, which led to the revolution of the Internet as a publishing medium to the Internet as a social environment [Lassila07]. Social networking is breaking boundaries in creating, representing and globally sharing information, and is providing new possibilities for human interactions. The second related trend is the shift of users from being consumers to becoming “*prosumers*” of content, i.e., both producers and consumers [ETP09], through social Web sites such as Facebook, or multimedia Web sites such as YouTube. The third trend is multi-media streaming, mostly in High Definition (HD). YouTube and Dailymotion are two examples of the many Web sites providing online videos that users can watch in high quality. It is further worth noticing that CISCO analysis predicts that 4.2 out of 6.3 exabytes crossing the mobile network per month will be due to video in 2015 [CISCO11]. 3D media is also expected to be a significant part of the Future Internet and is already being produced by large production companies. The transition towards a 3D-enabled Internet should include 3D Web browsers, 3D virtualization of users, and suitable intuitive interfaces for them to interact with each other, with information and with 3D media [FMI10, Kortuem09, Toma09, Eurescom08]. Another important evolution resulting from multimedia streaming, which is still at a relatively early stage, is the support of not only text-based search but also multi-media multi-model search over the Internet [FMI10, Stuckmann09].

The above will lead to increasingly heterogeneous data to be shared over the Internet, given the different formats of data to be produced by machines and users, and different formats resulting from content mashups [Schonwalder09].

It should further be noted that Web 3.0 technologies are expected to allow applications to perform intelligent processing of data and derive knowledge from information based on semantic tagging [ETP09]. This evolution, along with the increasing number of Internet users (foreseen to reach 7 billion by 2020) [Blackman10], will be key factors in an unprecedented increase of the amount of content to exchange.

Overall, the above trends shape the evolution towards the *Internet of Content* defined in [Daras09] as:

“Any type and volume of media. Content may be pre-recorded, cached or live, static or dynamic, monolithic or modular. Content may be combined, mixed or aggregated to generate new content and Media. It may vary from a few bits (e.g., the temperature that a sensor has measured) to interactive multi-media sessions and immersive complex and multi-dimensional virtual/real worlds’ representations”.

As a result, the Internet of Content should support applications that do not only seamlessly provide, but also process large quantities of information and support its propagation on both a local and a global-scale, while empowering more and more users to become prosumers of content. In fact, the Internet of Content provides location-independent access to objects and focuses on supporting interactions related to creation, distribution, and management of content [eMobility07, ETP09]. The Internet of Content should support bulk data transfers, real time data transmission and seamless information delivery. In [FIArch11], the authors state that Schmidt, the CEO of Google, estimated the size of information over the Internet to be 5 million Terabytes in 2005, of which Google had indexed only 200 terabytes, i.e., only 0.004%. Considering further that Internet content is expected to increase to 990 exabytes before 2012 [Theodore09], this illustrates the ultra-large scale envisioned for the Future Internet.

Internet of Services

Nowadays, service-oriented architectures are getting adopted to support the creation of services based on distributed components, along with Semantic Web technologies used to enhance service discovery and composition. However, services are still used in isolation from each other, with small-scale service compositions only. No Internet-scale service composition or service-oriented computing has been reached yet [ETP09]. Nevertheless, significant advancements can be identified with respect to the latter, mostly with the rise of Cloud computing, which aims at extending grid computing, distributed computing and parallel computing to provide resources at large as Internet services based on the virtualization of resources [Zhang10]. As in the case of content, another trend is user empowerment, allowing users to become prosumers of services and allowing them to design

their own service orchestrations. This has been made possible by the emergence of service mashup technologies, semantic description languages and service creation platforms [Schonwalder09].

In general, the above are the developing areas that will shape the *Internet of Services*, which is defined in [ETP09] as:

"An umbrella term to describe several interacting phenomena that will shape the future of how services are provided and operated on the Internet. The Internet of Services also comprises the various sets of Internet Applications including pervasive/immersive/ambient, industrial/manufacturing, vehicular/logistics, financial/ePayment/eBusiness, power network control/eEnergy, eHealth, and eGovernment applications."

Summarizing, the Internet of Services is expected to support an exponentially increasing number of context-aware services, i.e., services that are aware of their environment, their surrounding or their current situation. It will provide complex computation resources, and software functionalities as services [ETP09]. The Internet of Services shall also allow users to create any type of business, regardless of its size and domain. It shall handle large-scale virtualization and dynamic composition of services as well [Eurescom08]. The resulting Internet will be highly heterogeneous, in part due to service mashups along with the diversity of user-provided services and service application domains.

Internet of Things

Several physical things are already integrated in today's Internet. RFID tags, tiny sensors, mobile devices with embedded sensors and networked embedded devices are being largely produced and integrated within the Internet. Still, the Internet currently mostly includes a collection of uniform devices that will become increasingly heterogeneous as we move towards the future [Carnot11, CASAGRAS09, IOT08]. The Internet of Things is envisioned to include robots, smart buildings, smart fabric, smart books and so on, all designed to facilitate our daily lives [ETP09]. This implies a transition towards an increasing number of intelligent objects producing large amounts of data. Objects will be heterogeneous, as they will have different functionalities, different application fields, and different technologies [CASAGRAS09]. The transition towards the Future Internet of Things will be characterized by several technological evolutions: objects will evolve from connecting to networking to reach full intelligence; energy will shift from reduced consumption by low power chipsets to better batteries to reach energy harvesting capacities; devices will shift from small cheap tags and sensors to having better memories and sensing capacities at ultra-high speed and providing new physical effects; usage will shift from RFIDs in certain domains to global applications to finally reach unified networks that connect people, things, services and integrate industries. The transition will also be characterized by ground breaking technological advancements that will lead to nano-power processing units, wireless power, biodegradable devices and other advancements [IOT08].

As defined by the GASAGRAS³ project [GASAGRAS09] that is part of the EC FP7 ICT initiative towards the Future Internet, the *Internet of Things* is:

"A global network infrastructure, linking physical and virtual objects through the exploitation of data capture and communication capabilities. This infrastructure includes existing and evolving Internet and network developments. It will offer specific object-identification, sensor and connection capability as the basis for the development of independent cooperative services and applications. These will be characterized by a high degree of autonomous data capture, event transfer, network connectivity and interoperability."

Indeed, interconnected objects having each a unique way of identification and a capacity of addressing each other will characterize the Internet of Things. Objects shall have the capabilities to verify their identities, sense their environment, reason and learn, exchange and process information according to predefined schemas [ETP09, Thompson05]. Technologies supporting object-to-object communication interfaces and real-time localization are expected to be widely used within the Internet of Things [Toma09, ETP09].

The overall Future Internet:

All the aforementioned components of the Future Internet, i.e., Internet of Content, Services and Things, will cooperate in order to fulfill the requirements of the Future Internet users. On the one

³ CASAGRAS stands for "Coordination and Support Action for Global RFID-related Activities and Standardization"

hand, services that are aware of their physical surrounding and real world objects that provide services are expected to rise with the cooperation of the Internet of Services and the Internet of Things [Toma09]. On the other hand, the Internet of Content will cooperate with the Internet of Services and Things to provide users with content acquired through Services and/or Things.

2.1.2. Cross-domain Challenges of the Future Internet

On-going projects and research initiatives are thoroughly analysing the challenges and requirements of the Future Internet, as they constitute the key drivers for its design. Overall, the Future Internet is raising significant challenges over the computing and networking environments since it magnifies the features of the already challenging today's Internet, as summarized in Table 1. The following in particular surveys the main challenges posed by the Future Internet, i.e., *scalability*, *heterogeneity*, *mobility*, *awareness & adaptability*, and *security, privacy & trust*, which derive from the Future Internet constituents discussed in the previous section.

Challenges	Today's Internet	Towards the Future Internet
Scalability	1 billion Personal Computers (2008 ⁴), 647 million smartphones (2010) [CISCO11]	1.78 billion Personal Computers (2013), 1.82 billion smartphones (2013 ⁵)
	5 million terabytes of data (2005) [FIArch11]	990 exabytes of data (end of 2012) [Theodore09]
	10 ⁴ services (2007) [eMobility07]	Billions of services [TFI10]
	10 billion terminals (2010) [Hirabaru08]	100 billion terminals (2015) [Hirabaru08]
	Consumer Internet traffic of 12.684 exabytes/month (2010) [CISCO10]	Consumer Internet traffic of 42.070 exabytes/month (2014) [CISCO10]
Heterogeneity	Islands of interconnected objects	Internet-scale connection of highly heterogeneous objects (vehicles, sensors, mobiles devices, home appliances, etc.) [IOT08]
	Emergence of heterogeneous services provided on the Cloud such as Software as a Service (e.g., Google apps) or Storage services (e.g., Amazon service) [Zhang10]	Cloud computing enabling to provide everything as services, spanning different business and technical domains
	Service/content mashups leading to the provision of new, diverse services by <i>prosumers</i>	Global-scale services/content mashups creating new services/content with different types and formats
Mobility	Mostly (mobile) IPv4, which suffers from scalability issues etc.; even IPv6 has issues in mobile situations (e.g., due to the use of home agents/addresses) [ETP09]	Global-scale mobile Internet that requires revisiting communication/routing solutions [eMobility07]
	Wide-spread usage of smart mobile devices with limited resources (2 billion users)	Global-scale usage of smarter mobile devices with ever-growing resource needs
Awareness & Adaptability	Ad hoc solutions to network, content & service adaptation	Large scale content sharing, service provisioning, mobile connectivity that require autonomic adaptation and therefore awareness of content, networks and services [ETP09]
Security, Privacy & Trust	Safety and security requirements still an issue for today's Internet	Integrating real world objects, more users, more information, more services in the Internet intensifies the necessity for safety and security solutions

Table 1: The Future Internet Challenges

Scalable Internet

Based on our survey of the Future Internet literature, we identify *scalability* to be the most critical cross-domain challenge for the Future Internet to overcome. Indeed, the Internets of Content, Services and Things will face scalability issues with the increasing number, size and quality of their

⁴ <http://www.gartner.com/it/page.jsp?id=703807>

⁵ <http://www.gartner.com/it/page.jsp?id=1278413>

networked entities, which is further exacerbated by the empowerment of users [ETP09, Stuckmann09]. For instance, an important challenge within the Internet of Things is to handle the large amount of information provided by things and then to provide useful services given this information [ETP09]. We can also list the following challenges as part of the Future Internet's ultra-large scale:

- *Discovery and storage*: The large amount and size of content and the large amount of data provided by sensors will lead to higher storage requirements, in addition to challenges related to the discovery, search and retrieval of data [Toma09]. Similar challenges emerge for services as the order of services available on the Internet is expected to increase from 10^4 to billions in the future [eMobility07, TFI10].
- *Processing and data streams*: Another scale-related challenge is the processing of sensor data streams and the in-network processing of sensor data in addition to seamlessly streaming large amounts of multimedia data, probably 3D or high-quality data, which will be shared on the Future Internet [Stuckmann09, ETP09]. Dynamic composition of services will also be challenging with billions of heterogeneous services expected to be available.
- *Naming, addressing, routing and cross-layer communication*: As the number of systems and applications increases, the size of routing tables increases and efficient/stable/robust routing becomes more challenging. Providing a routing infrastructure that can handle scalability, and addressing spaces that can handle the overload over IP addressing space is a challenge that should be treated as high priority [ETP09]. The eMobility group argues that the IPv6 standard allowing the expansion of the address pool is not enough. It will also be a challenge to find a solution for combining existing address schemes with the ones to be developed for the Future Internet [eMobility07]. Cross-layer communication is not systematically considered as a core requirement by Future Internet projects. Nonetheless, its challenges should not be ignored, as there is a need for a right cross-layer information delivery without raising scalability and security issues [eMobility07, Jain06].

Heterogeneous Internet

The Future Internet will be heterogeneous in many dimensions, starting from physical objects, to networks, services and content, which present a significant challenge for sustaining the Future Internet vision [ETP09]. Therefore, semantic technologies, shared standards and mediation components are required to assure interoperability of heterogeneous components such as things, sensors, or networks over which content should be transferred without noticeable interruptions [Toma09].

Challenges in sustaining composition/integration in the Future Internet are directly related to its heterogeneity. Indeed, the content composition challenge, which may be considered scale-related, is mainly due to the increasing formats and types of content shared over the Internet. Furthermore, services and dynamic service compositions are expected to be increasingly heterogeneous with different business landscapes and different service abstractions. The emergence of Cloud computing providing diverse resources as services also contributes to increasing the heterogeneity of the Internet (e.g., with Software as service – Google apps; storage services – Amazon; etc.) [Zhang10].

Mobile Internet

The Internet was initially designed for fixed usage and mobility was the exception. However, this is no longer the case. Indeed, it is foreseen that two out of three billion hosts connected to the Internet will be mobile or other types of handheld devices by the end of 2011 [Stuckmann09]. Therefore, mobility should be natively integrated in the design of the Future Internet. A challenge that will rise lies in the design of a transmission protocol for a mobile wireless world, decoupling the network and mobile host identifiers from the TCP connection identifier [ETP09].

Surely, several mobility-related challenges should be addressed. Among those challenges lie: managing mobility of hosts, overcoming resource limitations (energy, processing power, etc.), overcoming intermittent connectivity to cope with transmission loss and then reacting appropriately, decoupling information from host and resolving the issues of Mobile IP (limited scalability) and IPv6 in mobile environments (e.g., due to the use of home agents/addresses). In fact, considerable efforts have been made to overcome the resource limitation challenges. Suggested solutions include adopting stand-by modes and duty cycles to save energy, or using Cloud computing to save energy

of processing resources by uploading complex computations to the Cloud [Kumar10]. To our knowledge, despite the attempts to solve most of the challenges above, no existing solution supports both a fully mobile Internet that does not rely on fixed nodes and handle the anticipated ultra-large scale usage of mobile nodes. As an illustration of the foreseen scale, there were 647 million smartphones in 2010 while it is expected to have 1.82 billion smartphones and phones equipped with Web access by 2013⁶.

Aware and Adaptive Internet

Awareness and related *Adaptability* are common requirements for sustaining the Future Internet, be it at the service, content or thing level. They are the source of several issues that should not be neglected. Challenges include: adapting the Web by and for users based on their context, adapting the network performance to the characteristics of shared media and vice versa, providing personalized content and media to users, providing context-aware dynamic services, and personalization of services [ETP09, Stuckmann09, Toma09]. Further challenges associated with enabling an aware and adaptive Internet can be represented in terms of autonomous management. Indeed, management, as perceived by the Future Internet vision, falls within the “aware Internet” challenge given that systems, networks, things and services in the Future Internet shall be able to self-manage (i.e., self-configure, self-heal, self-optimize and self-protect). Such a requirement is mostly due to the increasing number of services and underlying technologies that will increase network complexity to levels where human management will become infeasible [eMobility07]. Challenges further include: the management of virtual resources, cross-domain management, orchestration and dynamic deployment of management functions, in addition to addressing and routing information management [ETP09, Toma09]. Moreover, the rise of a global service delivery platform highlights the necessity to determine whom and how to manage service delivery on a global-scale [Toma09].

Safe Internet

Trust, *privacy* and *security* are sensitive cross-domain issues that the current Internet is facing, and these remain as sensitive challenges for the Future Internet. With the global-scale communications and exchange of information, users’ mobility and the limited resources their devices may have, the Future Internet’s “awareness” of users, their data, and their surroundings, it becomes more critical to find appropriate solutions that will protect users. By protecting users, we mean preserving their privacy, protecting them from attacks and ensuring that they can trust some service or person on another side of the globe. Moreover, the current security mechanisms will be unfit in the mobile world where users access their Internet *via* devices with limited resources. Therefore, security should be integrated in the architecture from the early design phases [eMobility07, ETP09]. As for privacy, authors in [ETP09] distinguish between two privacy issues: data privacy (related to users’ private data and information) and location privacy. Universal solutions shall be agreed upon to deal with the above-mentioned issues. However, those solutions do not exist yet, especially when it comes to presenting and managing identity data. Some technologies have been proposed such as “generic identifiers of electronic objects” or “real world object identifiers”. While those technologies do not provide an overall approach to handle the identity issue, virtualization can be a potential solution to handle a distributed management of identities [Toma09]. Further, appropriate governance and policies definition constitute key enablers for a safe Future Internet.

The question “Who will govern the Future Internet?” represents the main challenge related to the Future Internet’s governance. In fact, Governance is a cross-domain issue that requires not only technical but also legal and social involvements as well [Toma09, Stuckmann09, eMobility07]. Regarding policy management, there is a need to develop a universal language for policy management and digital licensing, in addition to a natural language to control devices [Kortuem10]. Moreover, data receivers should be able to specify policies in order to have control over packets they receive and the traffic coming their way [Jain06]. Overall, Future Internet systems should be self-governed and determine who has the right to take decisions, what policies to use and how to enforce those policies [ETP09].

⁶ <http://www.gartner.com/it/page.jsp?id=1278413>

2.1.3. CHOReOS Focus on the Future Internet of Services and (Smart) Things

The service-centric perspective of the Future Internet is acknowledged to play a paramount role [ETP09]. In fact, the *Internet of Services* will offer services for everyone and everything, across business sectors and areas of life. This will be achieved through so-called “service delivery platforms” including service-oriented architectures, Web 2.0-style interaction, semantically enabled processing, business models and deployment models.

Moreover, the Future Internet will go beyond today's traditional boundaries of the virtual world by being linked to the real world, hence leading to *the world of things and services*. The *Internet of Things*, enabling new services to connect the real life to virtual environments, will grow gradually through deployment of new applications (e.g., RFID product tracing, NFC payment devices, e-domestic appliances). As already discussed in the previous section and pointed out in [TFI10], “*progressively, as more real-life elements become connected, new services are emerging (e.g., mash-up applications), requiring appropriate levels of interface standardization and interoperability, of dynamic configuration capability, and an increase level of trust and associated information security supporting person privacy*”. In the context of the Internet of Things, the development of applications is expected to come along with a new range of user-centric services, based on the interaction of day-to-day processes with the network. The delivery of those services will be commonly seamless for users, requiring no specific interaction with them. The business model for the delivery of those services will require the interaction and collaboration of several organizations. Another challenge concerns also the network infrastructure. In particular, because of the massive increase of new entry points into the network, important aspects like scalability, communication protocols and middleware have to be properly taken into account.

It is sustaining service composition in the envisioned world of Services and (Smart) Things that CHOReOS concentrates on. Regarding the latter, we focus more particularly on the networking of *smart* resources including computing-enabled sensors & actuators and wireless computing devices such as smartphones, as opposed to merely passive, e.g., RFID-enabled, things. On the other hand, the dimension of “Internet of Content”, with the exception of machine-produced data, which is an integral part of the Internet of Things, is considered to be beyond the scope of CHOReOS, although possibly encompassed in the development of specific applications. Hence, by concentrating on enabling service composition in the Future Internet, CHOReOS investigates solutions at the middleware and above layers, following a service-oriented architecture perspective. As for lower-level layers, the required evolution will be based on solutions developed elsewhere.

CHOReOS shall then address the challenges listed in Section 2.1.2, which are associated with sustaining the Future Internet of Services and Smart Things in the specific context of service choreographies, i.e.: extreme scalability, heterogeneity, mobility, awareness & adaptability, and trust, security & privacy. The next section further refines the foreseen evolution of the Internet of Services and Smart Things, based on empirical studies about the evolution of service population and diversity that have been published over the last 10 years.

2.2. Evolution towards the Future Internet of Services and Smart Things

The growing richness of the Future Internet, concerning the services and smart things that will be available for use, is certain. On the positive side, this richness implies the availability of a large amount of design options that could be exploited by the developers of service-oriented software. On the negative side, this growing richness implies a growing complexity for the tasks that should be performed by the developers towards exploiting all these available design options, as well as for the runtime infrastructure.

In the remainder of this section, we attempt to characterize the growing plenitude of services and smart things in the Future Internet. To this end, we exploit various empirical studies that have taken snapshots of the available services in publicly accessible sources all over the Web [KIMR04, FANK05, LILZLXS07, AL-MASRIM08, SF09] and the Future Internet expectations documented in [ETP09]. Specifically, the empirical studies investigate the services that are available in standardized UDDI Business Registries (UBRs) [UDDI03]. Moreover, some of the studies that account for further types of sources for publishing services have also emerged recently, e.g., service portals or directories. The gradually widening of the spectrum of service sources is an indicator of the increasing number of providers that publish services, without adhering to the UDDI standard.

To characterize the plenitude of services and smart things in the Future Internet, we consider three different aspects in what follows: their population, their complexity and their functional diversity.

2.2.1. Population of Services & Smart Things

Regarding the current Internet, a rough comparison of the numbers of retrieved services reported in [KIMR04, FANK05, LILZLS07, AL-MASRIM08, SF09], shows a significant growth of the population of services that can be exploited by the developers of service-oriented software. This growth is even more noticeable in the recent years as depicted in Figure 1. The reason for this growth is twofold. First, as the years go by, we have more services published by service providers. Second, as the years go by, we have better search engines. In particular, all the studies apart from the latest two actually find a lower bound of the actual amount of available services. This is because they assume that all services are registered in the widely known publicly accessible sources. However, in practice, there are also services that are not registered in such sources and consequently, they are not traced by these studies. On the other hand, the latest two studies perform a more thorough crawling over the Web.

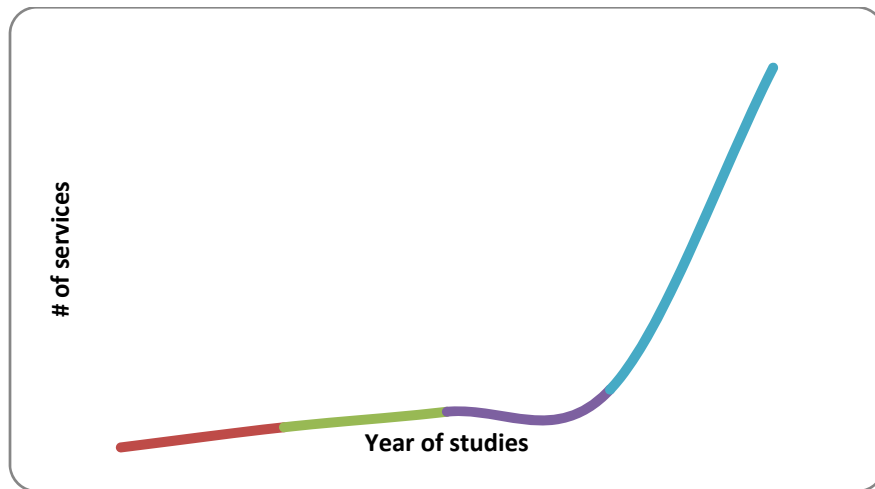


Figure 1: The Populations of Services observed from 2004 until now

Getting into the details of the studies, the earliest one found about 1000 services that were available on the Web in November 2003 [KIMR04]. Approximately, one year later, in June 2004, 2432 Web services were collected from all the widely known service sources of this period (Bindingpoint⁷, Salcentral⁸, Xmethod⁹, Webservicecx¹⁰, Webservicelist¹¹), revealing a noticeable 143% growth of the services that can be exploited by developers [FANK05]. In December 2006, the number of services that can be exploited by the developers keeps growing [LILZLS07]. In particular, 3522 Web services were found in the widely known service sources of this period (Bindingpoint, Xmethod, Strikeiron¹², Woogle¹³, Esynaps¹⁴, Webservicecx¹⁵) [LILZLS07]. In October 2007, 5077 Web services were retrieved by crawling the Web using the Google and the Yahoo search engines, and querying UBRs and widely known portals (e.g., Webservicelist, RemoteMethods¹⁶, WSIndex¹⁷, XMethods¹⁸) [AL-MASRIM08]. Finally, from December 2009 until now, the Service-Finder¹⁹ portal provides information for more than 27800 Web services. So, roughly from 2007 to 2009 [SF09], there is a significant 447% growth of the number of services that can be exploited towards the development of service-oriented

⁷ www.bindingpoint.com

⁸ www.salcentral.com

⁹ www.xmethod.com

¹⁰ www.webservicex.com

¹¹ www.webservicelist.com

¹² www.strikeiron.com

¹³ www.cs.washington.edu/woogle

¹⁴ www.esynaps.com

¹⁵ www.webservicex.net

¹⁶ www.remotemethods.com

¹⁷ www.wsindex.org

¹⁸ www.xmethods.net

¹⁹ www.service-finder.eu

software. Except for the development of new services, another interesting phenomenon is the upgrade of the already published ones. According to [LILZLXS07], the percentage of the upgraded services reaches 100% for each year from 2003 to 2006. These elements show that providers release new versions of their services quite fast. These versions aim at satisfying even more requirements covering a wider spectrum of consumers and business domains.

Concerning the near future, the growth of the Internet of Services & Smart Things is expected to be much more impressive, primarily due to the availability of service-enabled things. As documented in [ETP09], from 2010 till 2015, the population of available service-enabled things shall grow from millions to billions. Moreover, beyond 2015, the aforementioned population is expected to scale up to trillions of service-enabled things.

2.2.2. Complexity of Services & Smart Things

Certain of the empirical studies that took place from 2004 until now investigate the complexity of the retrieved services. In more details, they perform a quantitative evaluation of the complexity based on two metrics:

- The number of individual operations in a service (functions that the service offers to its consumers);
- The overall size of the service description.

The rationale behind these metrics is that services with many operations offer greater opportunities for composing them. On the other hand, the service description size depends on the number of parameters for each operation, and the structure of the XML type [XML09] for each parameter. Moreover, the description size may increase because of the textual information inside the description that explains the functionality of the operations.

Concerning the first metric, on average, the number of service operations is 5.5 [FANK05, LILZLXS07]. However, the percentage of complex services (i.e., services with more than 20 operations) is growing [FANK05, LILZLXS07].

Regarding the second complexity metric, in 2004, the percentage of service descriptions with size greater than 100Kbytes is 2.6% [FANK05]. In 2006, this percentage is 5% [LILZLXS07] whereas, one year later, this percentage is 10% [AL-MASRIM08].

Therefore, we can conclude that developers may exploit not only more services, but also more complex services, that provide more operations and/or more complex data types. To this end, the assistance that is provided to them is not sufficient. This becomes evident from the quality of service descriptions, which has been evaluated in [FANK05] and [LILZLXS07]. In particular, these two studies evaluated the amount and the sufficiency of the textual information included in service descriptions. The documentation within the service description provides a summary of the service functionality along with the meaning of each operation. In 2004, nearly 80% of the services have documentation of less than 50 words and almost half of the services do not have any documentation [FANK05]. Two years later, in 2006, 67% of services have documentation of less than 50 words and 27% of them do not have any documentation at all [LILZLXS07]. Hence, although there is a tendency for better documenting services, it still seems that the providers are not very much concerned about this issue.

In the future, the complexity of available services shall further increase. In particular, according to [ETP09], from 2010 till 2015, new services shall emerge, realized in terms of single domain compositions of base-level services offered by service-enabled things, while, the expectations beyond 2015 include even more advanced services composed of multi/cross-domain base-level services provided by service-enabled things.

2.2.3. Functional Diversity of Services and Smart Things

The evolution of the functional diversity of available services is investigated in [FANK05, LILZLXS07, SF09]. These works classify services according to their application domain. The different domains are organized roughly into 9 categories. A stimulating point is the evolution of service distribution into the categories during the last years. In early years, the majority of services belonged to limited categories. For instance, in 2006, 61.4% of services offered data lookup functionalities, 8.4% of the services provided data conversion functionalities, 7.8% of the services offered development tools functionalities, and 7.6% of the services provided communication functionalities [LILZLXS07]. On the

other hand, in 2009, according to the classification provided by the Service-Finder²⁰ portal, 37% of the services offered science-related (genetics and mathematics) functionalities, 27% of the services provided multimedia functionalities and 17% of the services provided shopping functionalities. The population of the remaining top 6 categories ranges from 1% to 6% of the overall amount of services as depicted in Figure 2. Consequently, we observe that more services become available in multiple categories showing a tendency towards a more uniform distribution of services into the categories than that in 2006.

Hence, nowadays, the functionality of services varies in a greater extent than the early years indicating the proliferation of service development in more application domains.

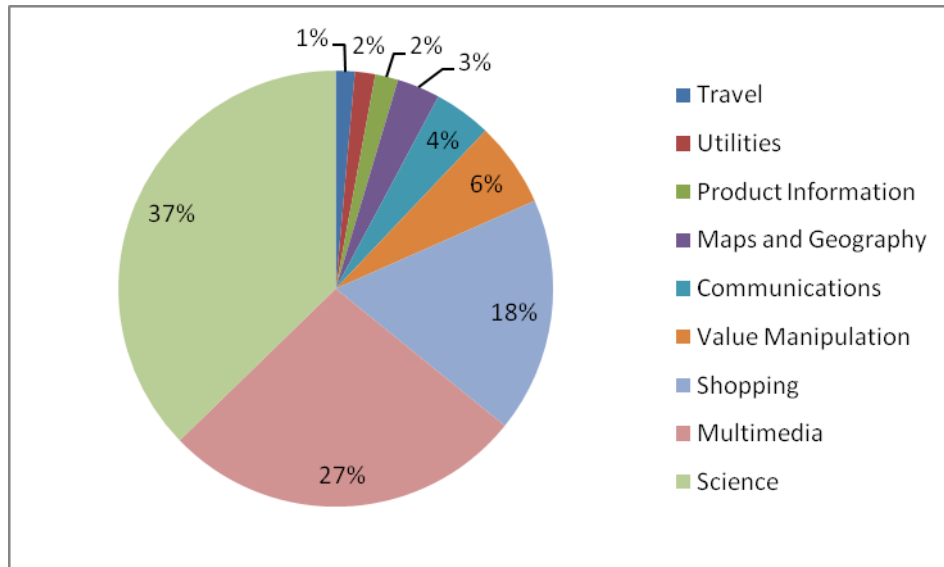


Figure 2: The Functional Diversity of Services in 2009

Concerning the future, the diversity of available services is expected to further grow with the availability of service-enabled things that would facilitate the development of more complex service-oriented software such as the social and enterprise applications that will emerge from 2010 to 2015, the goal-oriented distributed things-to-things collaboration environments that will be available from 2015 till 2020 and finally the intelligent things-to-humans collaboration environments that will emerge beyond 2020 [ETP09].

2.2.4. Concluding Remarks

As surveyed above, the Internet of Services and Smart Things is becoming a reality with a population of services, including service-enabled things, called to evolve at a very fast pace. However, assisting the developers in leveraging such a plenitude of services to provision new services through service composition raises tremendous challenges. Obviously, core challenges follow from the challenges arising for the Future Internet discussed in Section 2.1. However, within CHOReOS, by focusing on sustaining the choreography of services composing the Future Internet, we are more specifically concerned with addressing those challenges in relation with core building blocks of the service-oriented architecture, i.e., service discovery, access and composition. In addition, we will investigate those challenges in close relation with requirements from representative case studies, among which the DynaRoute case study analysed in the next section.

2.3. Future Internet Requirements from the DynaRoute Use Case

CHOReOS investigates the challenges and requirements of the Future Internet both in a top-down, mostly based on the literature, and bottom-up fashion, based on concrete case studies. While the two previous sections have concentrated on the top-down analysis, this section concentrates on the bottom-up one, focusing on one of the case studies to be experimented with within CHOReOS, i.e., the DynaRoute use-case.

The DynaRoute use-case (see CHOReOS DoW-WP8) depicts a situation of a person following a predefined “itinerary” (i.e., a scheduled sequence of activities) on her way from her hotel to the

²⁰ <http://demo.service-finder.eu/search>

airport. The DynaRoute activities involve the dynamic synthesis and adaptation of choreographies, which are controlled and modified on-the-fly, based on external triggers and condition changes. In more detail, the DynaRoute scenario involves the actors listed in Table 2 below, as part of the seven choreographies summarized in Table 3.

Type of Actors	DynaRoute Actors
People	Travellers, friend(s), taxi drivers, porters
Things	Taxis, navigators, monuments, airline companies
Services	Services for navigation, relaying traffic information, store (sales) announcements, tour-guide, luggage transfer, departures timetable, etc.

Table 2: Actors of the DynaRoute Choreographies

Choreography	Choreographed Parties	Description
C1 Context-aware adaptive itinerary	Relevant networked services deployed (i) in the environment and (ii) on the users' handhelds (e.g., smart phones)	Computes adaptive itinerary for users according to their situation, their travel constraints, and environment.
C2 Co-taxi-ing	Customized networked services deployed (i) on the users' handhelds and (ii) in taxis	Tries to assign a subset of people waiting for taxis, to each available taxi.
C3 Distributed alert system	Customized networked services (i) provided by the airline companies and (ii) deployed on the users' handhelds	Implements push-based communication service between the airline companies and their customers.
C4 Context-aware, distributed ad hoc social networking	Socially-based networked services deployed on the users' handhelds	Supports proximity-based social networking.
C5 Context-aware, touristic guide	Networked services (i) embedded in the touristic area, and (ii) deployed on the users' handhelds	Offers touristic services that are adapted to match the tourists' interests and travel status. For example, touristic information may take the form of a short textual description, a longer video tour, or one full scale guided tour, depending on the agenda of tourists.
C6 Context-aware distributed, ad hoc traffic management	Traffic management services embedded in taxis	If a taxi identifies traffic jam or other problems (e.g., faulty traffic lights, accidents etc.), it can pass this information to other 'nearby' taxis. On a hop-by-hop basis the information is propagated to all, hence realizing a multi-hop ad hoc mobile sensor network dedicated to traffic management
C7 Context-aware, personalized airport ground service	Customized networked services (i) offered by the airport and (ii) deployed on the users' handhelds	The porters at the airport are able to get dynamically notified about the check-in counter of the traveler, the baggage drop, or any other changes to the traveler's flight.

Table 3: The DynaRoute Choreographies

Table 4 below summarises the DynaRoute Future Internet-related requirements, which represent a high-level overview of the specific requirements of DynaRoute in the context of the Future Internet challenges highlighted in Section 2.1. The DynaRoute requirements will be further analysed and elaborated in Deliverable D8.1 to become full system requirements.

Future Internet Challenges & Requirements		'DynaRoute' Specific Requirements
Scalability	Number, size and quality of networked entities	<p>C1 is a 1-to-many choreography, where 'many' is expected to be in the order of a few hundreds to thousands.</p> <p>C2 is between m taxis and n people where we expect the levels of taxis to be around a few hundreds. This also hints on the number of people awaiting taxi service, which, however, depends on time of the day.</p> <p>C3 is proportional to the number of passengers of an airline. This can vary from a few tens to a few hundreds depending on the number of flights to different destinations with the same airline.</p> <p>C4 is limited by the geographical area. Assuming that in a hot spot area, there are numerous socially-related people, this is at the levels of several tens.</p> <p>Regarding C5, the number of tourists at or nearby a monument is about a few thousands.</p> <p>For C6, the number of taxis, in a medium city, is of about a few thousands (C5).</p> <p>As for C7, the number of porters/passengers is in the order of hundreds.</p> <p>According to the above, the complexity of DynaRoute in terms of numbers of entities is relatively small at the Future Internet scale (a few thousands is the worst case). However, it is already challenging regarding the number of actors to be concurrently involved within a choreography.</p>
	Storage (scale and size of content and sensors' data)	The scale of data and their size for DynaRoute is proportional to the number of devices. As such, the information that needs to be kept and stored is not expected to be a major challenge, which fits CHOReOS as content management in the Future Internet is not part of the project focus.
	Discovery (search & retrieval)	Search in DynaRoute is overall challenging. For example, matching the various people in a certain geographical area, who are socially related, under constraints of time and itineraries can be quite challenging.
	Streaming large amounts of data	In DynaRoute, the biggest challenge in this respect is the one in C5 where high capacity multimedia streams need to be sent to different tourists, with different levels of Quality of Service (QoS) supported by their devices, different languages, etc.
	Addressing and Naming	Addressing is very much related to the naming of devices. If we assume IPv6 and due to the relatively low level of DynaRoute requirements when it comes to number of devices, it should suffice to cover the case study.

Interoperability	Interoperability of heterogeneous components	This is one of the biggest challenges and requirements from DynaRoute. Indeed, no one can expect that the devices in taxis, held by different people (travelers, porters, friends), and deployed at monuments, etc., will be the same and running matching protocols, from application down to network layers.
	Content composition & integration	Dynamic content integration is not one of the most important challenges for the DynaRoute case. C6 (information exchange among taxis) is one of the choreographies of the case study that will need such dynamic content integration. For example, multiple views of the same (or similar) phenomenon from different taxis and the interpolation to a final assessment are going to be needed.
	Dynamic service composition	DynaRoute heavily relies on dynamic composition of services, as can be inferred from the description of choreographies in Table 3.
Mobility	Native support/integration of mobility	Mobility is needed for almost all the DynaRoute choreographies.
	Routing (efficiency, stability, robustness)	Routing is an open research issue for mobile sensor ad hoc networks. In particular, the highly dynamic ad hoc network of taxis is raising a number of challenges.
	Limited resource mobile devices	Some devices, especially the mobile users' devices have very strong demands on resources and especially energy.
Awareness & Adaptability	Adaptability	Adaptability is a major requirement for DynaRoute. For instance, C1 needs to adapt to different circumstances and interact with other choreographies.
	Awareness	Awareness is also key to DynaRoute. Both context and self-awareness are needed. For example, awareness of other devices, of other people, of weather conditions, of shopping hours, and of traffic jams, are all key to the DynaRoute case study.
	Manageability (self *) *: awareness, healing, optimization, configuration, maintenance	As mobility increases and as people rely more and more on smart devices while being on the road and vulnerable, it is of paramount importance to have self-manageability, auto configuration, etc. Indeed, there will not be any IT support service for any of the actors (with the exception of monument) during the DynaRoute case.
	Monitoring and measurement	Monitoring of traffic, other devices, environment, etc. is of paramount importance to DynaRoute as many of the services rely on monitored information.
	Resilience against accidents & failures (fast recovery, reliability, incident detection and analysis)	Resilience and recovery are major requirements for DynaRoute. When having to deal with time-sensitive services and users being highly dependent on services, like DynaRoute does, resilience and fast recovery are considered essential. Levels of seconds should be considered as worst-case recovery times.

Security, Privacy & Trust	Privacy of users' data	<p>In DynaRoute, there is a lot of private data (location, itinerary, habits, friends, destination, etc.) exchanged and being exposed. As the data privacy regulations in Europe are quite strict, this is considered a major challenge for DynaRoute.</p> <p>The regulations of National DPAs²¹ and the directives and opinions of the EDPS²² must be adhered to.</p>
	Protect users from attacks	<p>This is a less important aspect for DynaRoute. It all depends on the level of the person(s) involved. If the people involved are heads of states or VIPs, then, indeed this can be a major requirement.</p> <p>But as DynaRoute is targeting the average person, this can be of a lesser importance.</p>
	Ensure trust between services and people	<p>Trust is major concern for all interaction-based services. DynaRoute is based on trust relationships between humans, which then have to be transformed into trusted relations between devices as well.</p>
	Identity management and preservation	<p>This is a major requirement. People should be able to be identified and given access to certain services only if they are allowed to.</p> <p>For example, it should not be possible to change the flight arrangements of a certain passenger.</p>
	Governance and policy management	<p>DynaRoute requires a distributed governance structure.</p>

Table 4: The DynaRoute Requirements

Our characterization of the Future Internet focusing on the Internet of Services and Things, as elicited in this chapter both in a top-down and bottom-up fashion, raises a considerable set of challenges for related system development targeted by CHOReOS. This calls for a precise and rich conceptual model, which will serve as foundation to the choreography-based Future Internet of Services and Things and especially to its supporting infrastructure. Before introducing the initial conceptual model for CHOReOS in Chapter 4, we provide in the next chapter an overview of the NEXOF Reference Architecture and embedded conceptual model, which will be used as a baseline for the definition of the CHOReOS conceptual model.

²¹ DPA: Data Protection Authority

²² EDPS: European Data Protection Supervisor

3. The NEXOF Reference Architecture

As already introduced, a conceptual model provides a high level common ground by capturing the relevant entities/concepts, and relationships among them, underlying a given class of systems.

For CHOReOS, the considered class of systems consists of choreography-based service-oriented systems in the Future Internet, and hence service-oriented systems realized as choreographies of services discovered in ULS networking environments. The highly-scalable choreographies that CHOReOS wants to investigate, as part of Future Internet scenarios, need to be understood and their key characteristics need to be identified by taking into account the state of the art on conceptual models and languages for service-oriented systems at large and service-oriented choreography in particular, in relation with the key features of the Future Internet.

As reported in [D1.1], a number of conceptual and architectural characterizations have been proposed in the literature for modeling/describing service-oriented systems: the Web Service Architecture [BHMN07], the CBDI Model [CBDI07], the OASIS Reference Model [OAS06], the EGA Reference Model [EGA106,EGA206], the TrustCoM Conceptual Models [SWHA05], the Service Component Architecture [BBBE07], the Service-oriented architecture Modeling Language [SoaML], the SeCSE Model [CDDD05,SeCSE] (to which the CHOReOS partner CITY contributed), and PLASTIC [PlasticWS,Plastic08] (where the CHOReOS partners INRIA, CNR-ISTY, VTRIP, and UDA had an involvement).

The most recent approach is the one adopted by the NEXOF Reference Architecture (NEXOF-RA)²³ project, which proposes a holistic approach that refines and enriches all the previous efforts in the area [SLC10]. Shortly, the NEXOF-RA project is the first step in the process of building the overall NESSI²⁴ Open Service Framework (NEXOF)²⁵, *the generic open platform for creating and delivering applications enabling the creation of service based ecosystems where service providers and third parties easily collaborate*. In particular, NEXOF-RA is a *pattern-based reference architecture* for SOA infrastructures within the Future Internet and, as part of it, a conceptual model specifies the first-level entities that constitute these infrastructures, the facilities they provide in support of service-oriented systems, as well as relationships among them.

The analysis of the NEXOF-RA has led us to conclude that it is a valuable work towards the establishment of a common reference architecture, and its related conceptual model, of the infrastructures supporting service-oriented systems in the Future Internet. However, the notions offered by NEXOF-RA do not address the specific facets of CHOReOS, and in particular do not consider the choreography of services as a first-class entity to build service-oriented systems within the Future Internet. Moreover, as indicated in Chapter 2, CHOReOS leverages the Internet of (Smart) Things domain, which is not currently considered by NEXOF-RA. Hence, in Chapter 4, by adopting the way the NEXOF-RA conceptual model is structured and presented, we introduce new CHOReOS-specific notions and, when needed, we take care of making explicit relationships with those already defined by NEXOF-RA. To this end, hereafter in this section, we provide an overview of the whole NEXOF-RA by showing figures borrowed from the official NEXOF-RA Web site and documents therein, so that the deliverable is self-contained.

3.1. The NEXOF Reference Architecture in a Nutshell

NEXOF-RA is “a set of instruments designed to help the construction of well-architected distributed SOA software infrastructures” within the Future Internet.

Special emphasis must here be given to the term “SOA infrastructures”. According to the NEXOF-RA view, any software system can be divided into the *infrastructure* and the *operative elements* that can be deployed on it (see left-hand side of Figure 3). From this standpoint, SOA introduces the notion of distributed SOA infrastructures and service-oriented applications that can be deployed and executed on it, in order to focus on business objectives. The rationale behind distinguishing between SOA infrastructure and service-oriented applications on it is that “SOA is *not a specific technology or predefined solution but rather a paradigm or architectural style that is used to improve the scalability*

²³ <http://www.nexof-ra.eu/>

²⁴ <http://www.nessi-europe.com/>

²⁵ http://www.nessi-europe.com/?Page=open_innov

and decentralization within distributed, heterogeneous, cross-business-domains IT environments". Specific characteristics of a service-oriented system that will actually be deployed are likely to be unique within each business domain and company. Therefore, it is not feasible to define a "one-in-all" SOA solution that fits all situations, rather, a SOA solution needs to be adapted to each individual context. However, each SOA solution is characterized by the usage of technologies and platforms (and hence infrastructures) that specifically support, e.g., the creation, execution, and delivery of services.

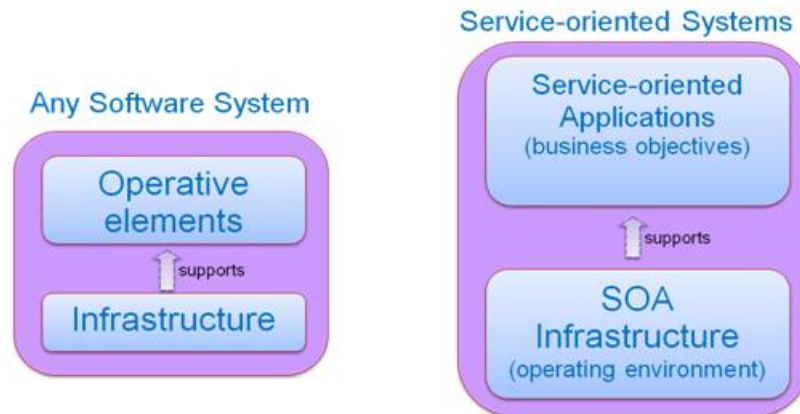


Figure 3: Organization Structure of Software Systems

Thus, the NEXOF-RA project team has chosen a domain- and technology-independent *pattern-based* approach to model SOA infrastructures capable of supporting different types and demands of service-oriented systems. As such, NEXOF-RA is in principle able to deal with often contradicting requirements of heterogeneous classes of systems, e.g., Software as a Service, Cloud Computing, Internet of Services, Internet of Things, Web 2.0/3.0. In line with the well known idea underlying architectural or design patterns, the NEXOF-RA pattern-based approach allows for deriving specific infrastructure architectures for specific classes of systems by means of a set of patterns that can be composed according to specific requirements. NEXOF-RA considers a pattern as a first class design element, which describes part of the infrastructure, detailing the pieces it is composed of, the role of these pieces, and how they integrate and interoperate. A pattern also provides guidelines for implementation.

As reported in [NEXOFRAS10], a set of patterns has been already developed for describing the characteristics of the following classes of service-oriented systems:

- Enterprise SOA,
- Internet of Services, and
- Cloud and Infrastructure as a Service.

The NEXOF-RA project team claims that the Internet of Things can easily be integrated into the NEXOF-RA as a new system of patterns, and this is planned as future work. Such a future direction is of interest for CHORéOS since, as already said in Chapter 2, one of the main objectives of CHORéOS is contributing to leveraging the Future Internet of Services and Smart Things. In particular, CHORéOS aims at offering an Integrated Development and Runtime Environment for decentralized choreography-based service-oriented systems built out of constituents of the Internet of Services and Smart Things.

As pointed out in [SLC10], the specification of a pattern system is not "per se" sufficient to provide a valuable reference architecture from which specific architectures can be derived. This is why, in addition to the pattern system, NEXOF-RA is composed of several parts that foster the instantiation into specific architectures. Referring to Figure 4, the top-level parts are:

- (1) The *Guidelines and Principles* [NEXOFAFP10],
- (2) The *Reference Specification* [NEXOFRAS10] containing the *Pattern Ensemble*, and
- (3) The *Reference Model* [NEXOFRM10], mainly containing the *Conceptual Model*,

which we further define below.

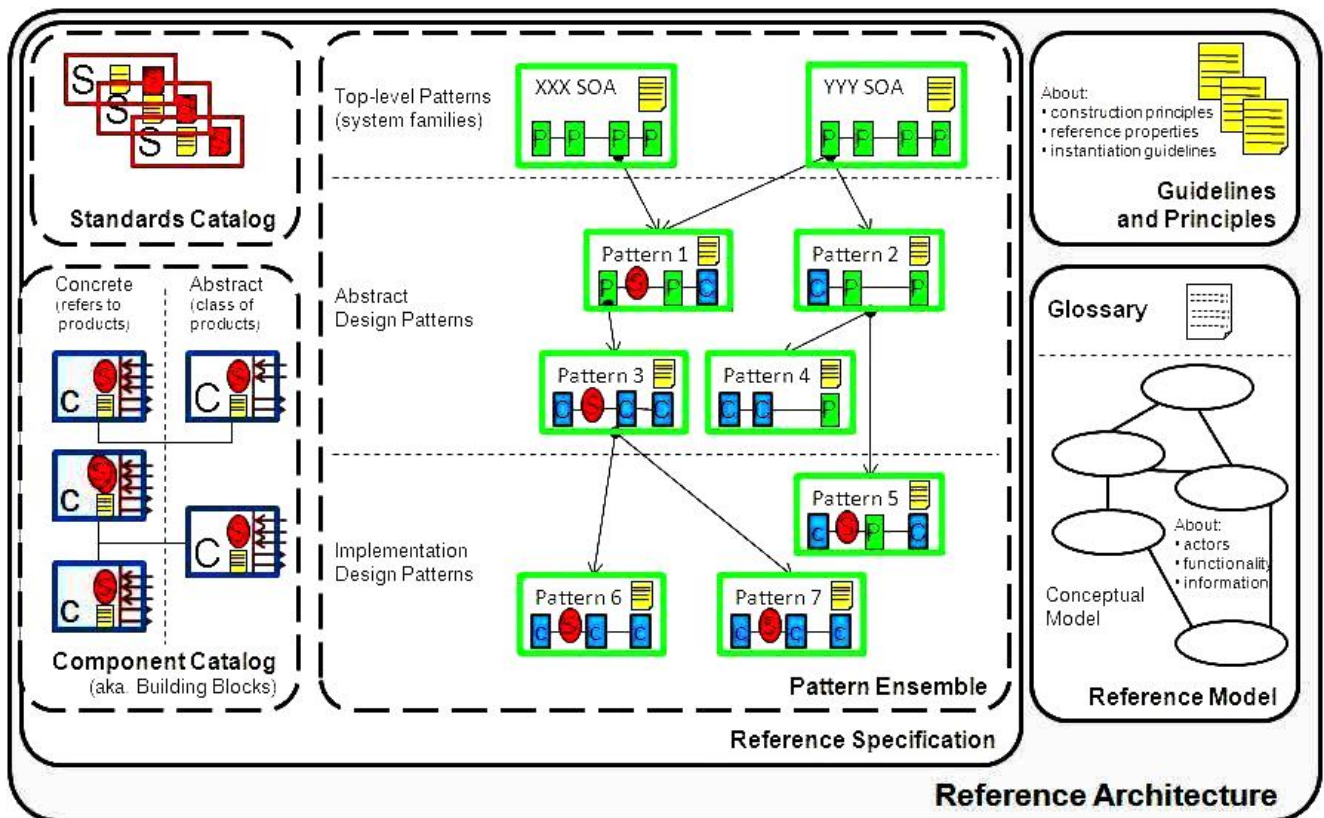


Figure 4: Structure of the NEXOF Reference Architecture

3.1.1. Guidelines and Principles

Guidelines are used to instantiate a specific system architecture according to its requirements, e.g., help on matching/selecting a certain pattern from a given requirements.

Principles concern the development and evolution and are used to create and to evolve the NEXOF-RA specification, e.g., fix rules and restrictions, describe the structure of the patterns and how new patterns can be added, allowed relationships among patterns.

3.1.2. Reference Specification

This part decomposes into the following sub-parts:

- The *Standards Catalog* describes standards and specifications referred to in the NEXOF reference architecture.
- The *Component Catalog* groups abstract descriptions of component products (e.g., a UDDI registry) and specifies the links to concrete software components that implement them (e.g., the jUDDI library). Each description refers to the standards it implements, the concepts it addresses, as well as its behavioral characteristics.
- The *Pattern Ensemble* describes patterns as first class design elements; they are called constructional-patterns and can be used in the design of a system as any other design element (class, module, component etc.). Patterns are organized according to three levels of abstractions:
 - *Top-level* patterns are architectural patterns that describe the fundamental structural organization schema of SOA platform families (Enterprise SOA, Internet of Services, Cloud and Infrastructure as a Service) in terms of predefined subsystems or components.
 - *Abstract* patterns are design patterns that refer to abstract components and provide a schema for refining the subsystems or components of a software system or the relationship between them.
 - *Implementation* patterns are low-level patterns that refer to at least one concrete component specific to a technology: standard, programming language, product. They

describe how to implement particular aspects of components or the relationships between them using the features of a given technology.

The patterns can be further classified according to the following categories:

- *Functional / Non-Functional Patterns*
- *Cross-Cutting / Non-Cross-Cutting Patterns.*

A *Functional Pattern* specifies an architectural solution to a software system by mainly considering the functionalities it must provide.

A *Cross-Cutting Pattern* can be either a functional pattern or a non-functional pattern that can be applied to other patterns. It provides guidelines to transform and enhance the architectural solution provided by another pattern. Thus, patterns that are not applicable to others are *Non-Cross-Cutting Patterns*.

A *Non-Functional Pattern* is a cross-cutting pattern that is designed to be applied to other specific patterns in order to improve quality attributes.

Since, understanding and using patterns individually does not help much, NEXOF-RA also specifies relationships between all the patterns that are produced. Thus, for each pattern, the relationships with other patterns are provided, so that the overall pattern system can be used to scale beyond point solutions, and thus to address larger and more sophisticated problem spaces [NEXOFAFP10].

3.1.3. Reference Model

The NEXOF-RA *Reference Model* defines the first-level entities that constitute the infrastructures underpinning heterogeneous service-oriented systems, as well as the relationships between them, and can be used to conceptually describe different system views by following the well-known approach of separating structure, behavior and functionality. The *Reference Model* part comprises the *Glossary* and the *Conceptual Model*.

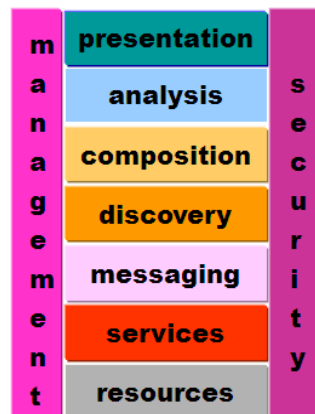


Figure 5: NEXOF-RA Concerns

The glossary contains NEXOF-RA-specific and Not-NEXOF-RA-specific terms. The *NEXOF-RA-specific terms* include all key terms for the NEXOF-RA scope and are organized according to the *NEXOF-RA layered concerns* (see Figure 5), where we find from the top to the bottom:

- **Presentation** concern: deals with the incorporation of people into the SOA equation,
- **Analysis** concern: deals with continuous process improvement,
- **Composition** concern: deals with the integration of services into business processes,
- **Discovery** concern: deals with reuse,
- **Messaging** concern: deals with communication and interaction with services,
- **Services** concern: addresses the underlying building blocks of SOA
- **Resources** concern: deals with SOA enactment.

We further identify the following NEXOF-RA orthogonal concerns:

- **Management** concern: deals with service level agreement and governance
- **Security** concern: deals with SOA reliability.

Since the overall NEXOF-RA is structured around the above nine concerns, the specification of the conceptual model is organized accordingly.

3.1.4. Reference Model vs Reference Specification

The *Reference Model* (and hence the *Conceptual Model* as part of it) plays a decisive role within the creation/evolution of the NEXOF-RA specifications itself as well as in the instantiation of NEXOF-compliant architectures. According to Figure 6, while the *Reference Model* specifies **what** are the facilities provided by any NEXOF-compliant infrastructure in support of the achievement of service-oriented systems business objectives, the *Reference Specification* describes **how** these facilities can be provided.

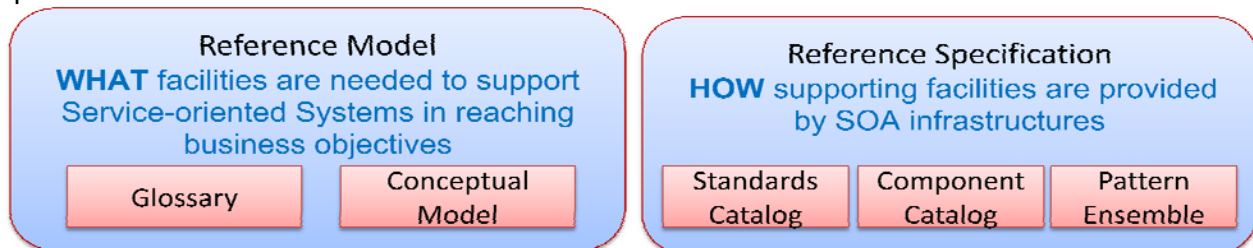


Figure 6: NEXOF-RA Reference Model vs NEXOF-RA Reference Specification

More specifically, within the *Reference Specification*, *Patterns* are designed for fulfilling the functionalities formalized by the *Conceptual Model* and are indirectly related to service-oriented systems' business requirements (see Figure 7). The *Conceptual Model* addresses and covers all the functionalities of a generic service-oriented system infrastructure platform. According to the objectives of a specific *Business Scenario*, (part of) these functionalities will be provided. Moreover, patterns are also designed to fulfil the *Non-Functional System Requirements* of such a service platform derived from the business scenario. Finally, patterns are validated by *Proof Of Concepts* (POCs). A POC is defined as a (set of) software artefact(s) used to validate some key patterns of the Reference Specification. In particular, the main focus of a POC is on the validation of patterns' claim about quality attributes.

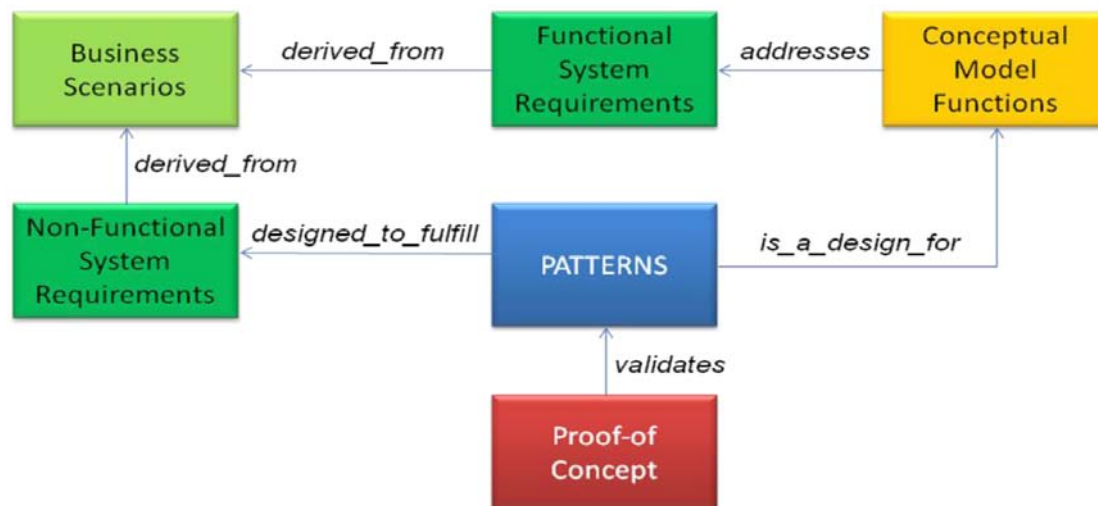


Figure 7: Relationships among Business Requirements, Conceptual Model, Architectural Patterns, and Proof-Of-Concept

Within the NEXOF project, quality aspects (such as high availability and scalability of SOA infrastructures) are not explicitly addressed in the NEXOF-RA conceptual model. In fact, according to the vision of the NEXOF-RA project team, quality attributes are inherent characteristics of specific architectural solutions and can be ensured by (composition of) different architectural patterns. Hence, they cannot be expressed on a conceptual level independently from specific solution aspects. That is, for each pattern, the NEXOF-RA specifies the influence that the usage of the pattern has on quality attributes.

The next section presents an overview of the NEXOF-RA *composition* concern that CHOReOS extends to provide a characterization of service choreographies. Moreover, as presented in Chapter 4), the CHOReOS conceptual model is called to extend all the NEXOF-RA concerns to address the specifics of Future Internet Choreographies.

3.2. Overview of the Composition Concern

As said, the NEXOF-RA conceptual model describes each concern by following the well-known approach of separating *functionality*, *structure*, and *behavior*. Functionalities are defined by use case diagrams, the structure is defined by class diagrams, and the behavior is defined by activity diagrams (data flows in the NEXOF-RA terminology). In particular, each functionality is described in detail in [NEXOFRM10] together with explicit dependencies with the system requirements and references to their sources (e.g., NEXOF-RA Investigation Team, Products, Standards, Books).

As an example, Figure 8 reports all the (sub)functionalities of the *Composition* concern. In particular, service-oriented infrastructures should support the **Creation** and **Execution** of composition of published services components. The creation encompasses the **Design** and **Implementation** of the composition process being specified. The composition process can be in turn published (i.e., **promoted**) as a composite service.

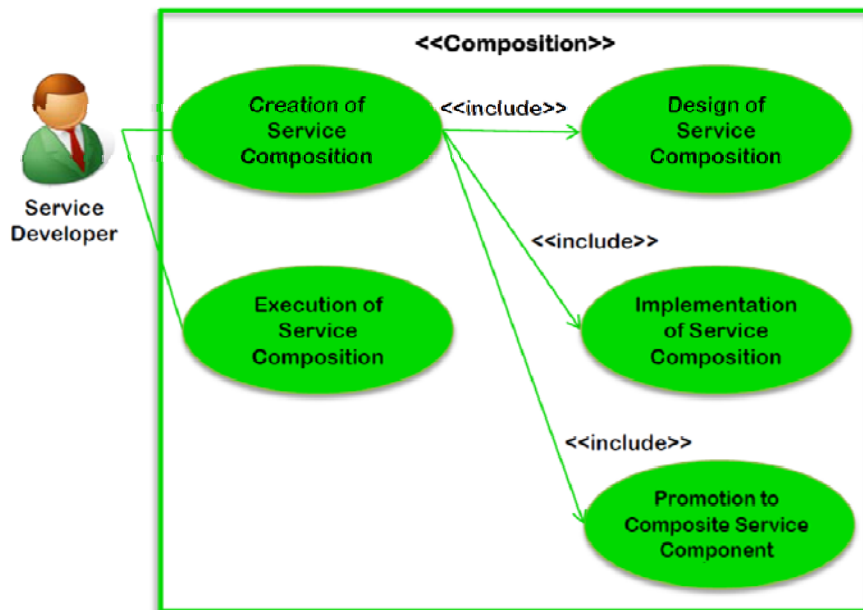


Figure 8: Functionalities of the *Composition* concern

The structure and the behavior views (i.e., the data flow) of the composition concern are shown in Figure 9 and Figure 10, respectively. Figure 9 and Figure 10 can be easily understood and they are briefly described here just to let the reader grasp the way the NEXOF-RA concerns are presented. Still, interested readers are referred to [NEXOFRM10] for more details.

A **Service Composition** is described by a **Service Composition Description**, i.e., either an **Orchestration** or a **Choreography** (see Figure 9). Based on existing **Service Descriptions**, the service composition description specifies the **Data Flow** and the **Control Flow** of the service composition. Implemented by an **Executable Workflow**, a service composition can also be exposed as a **Composite Service**, in turn implemented by a **Composite Service Component**.

The composition concern addresses the functionalities **Creation of Service Compositions** and **Execution of Service Composition**, which together allow handling service compositions (see Figure 10). Considering the requirements the composition has to fulfill and the descriptions of the required services discovered into the registry, developers can create (i.e., design and implement) service compositions and execute them. The implementation of a service composition leads to the creation of a **Composite Service Component** that, after being **deployed**, can be **run** to actually execute the created service composition. The **Execution of Service Compositions** is performed in the form of a **Executable Workflow** upon the event of **dispatching** a message that starts the invocation of the composition. The executable workflow is processed by giving place to a series of **Observable Actions** for the different services that are invoked within the process execution.

4. Initial Version of the CHOReOS Conceptual Model

As described in the CHOReOS DOW, one of the main goals of CHOReOS is to realize in WP5 an Integrated Development and Run-time Environment (IDRE) that, integrating the outcomes of the WP2-3-4 RTD work, constitutes a software infrastructure able to support, from development to (and during) run-time, choreography-based services in the Future Internet. As depicted in Figure 11, the *CHOReOS Conceptual Model* shall then highlight the key facilities to be supported by the *CHOReOS IDRE Architecture*.

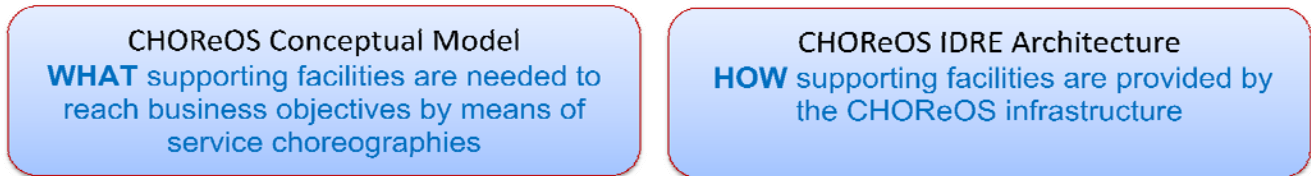


Figure 11: CHOReOS Conceptual Model vs CHOReOS IDRE Architecture

More precisely, independently from any implementation and technological details, the CHOReOS conceptual model describes on a conceptual level **what** are the facilities offered by the CHOReOS infrastructure, which are needed to support the achievement of business objectives by means of service choreographies. The CHOReOS IDRE architecture then specifies **how** the given facilities are provided by the CHOReOS infrastructure and hence by the IDRE.

Note that the relationship highlighted in Figure 6 for NEXOF-RA and the relationship in Figure 11 for CHOReOS must be considered at different levels of abstraction and generality. In fact, while the NEXOF conceptual model and the NEXOF reference architecture concern supporting facilities for general service-oriented systems, the CHOReOS conceptual model and the CHOReOS IDRE architecture concern supporting facilities for the class of service-oriented systems realized as choreographies of services. Moreover, the NEXOF reference architecture is an abstract specification that can be used as a reference for architecting specific and concrete SOA infrastructures; whereas, the CHOReOS IDRE architecture is per se a concrete specification of the SOA infrastructure offered by CHOReOS for supporting the design, development, enactment, and validation of choreography-based large scale services in the Future Internet, i.e., “CHOReOS choreographies”.

Overall, the definition of the CHOReOS conceptual model is strongly related to the activities defined in the CHOReOS Development Process Model, which is introduced in the CHOReOS DOW and sketched in Figure 12. These activities decompose into:

- First, as depicted in Figure 12, CHOReOS intends to support the systematic development of choreographies from their design to their actual enactment, i.e., execution. CHOReOS in particular investigates techniques and tools that emerge from MDE and generative programming research areas, and makes use of two orthogonal transformational approaches: (i) a top-down transformation process and (ii) a cross-cutting transformation process. The former will serve to refine the *i**-based *domain-expert requirements* specification into *analysis-technique-specific models* derived from a peer-style specification. The latter will serve to integrate the different *modeling/reasoning technologies* by passing from a technique-specific model to a different technique-specific model. This will bridge the gap between the various models that have to be used for choreography synthesis, analysis, validation, and implementation purposes.
- A distributed (large scale) *service base* (aka registry) will further manage information about available services offered by service providers. As deeply discussed in [D1.1], the service base organizes available services into functional and non-functional views. Each view is characterized by corresponding abstractions (functional/non-functional) and a set of available services that are represented by each abstraction.
- The approach envisioned for *the choreography synthesis* then starts from (i) the *peer-style specification* of the choreography and from (ii) the set of *services discovered* from the large scale service base. Input (i) is based on the functional and non-functional abstractions, and it comes from the refinement of the domain expert specification obtained by means of the top-down transformational process. Input (ii) comes from the exploitation of the service base

management mechanisms. That is, the synthesis process assumes that the services into the registry/base have been discovered so that they satisfy the local (to the service) functional and non-functional requirements that have been specified for the choreography and, hence, can be considered as potential candidates to participate in the global choreography process. The choreography synthesis produces (possibly with partial human intervention/inspection) the abstract proxies/adaptors that, supported by the CHOReOS service-oriented middleware to be developed in WP3, distributedly support the enactment of choreographies.

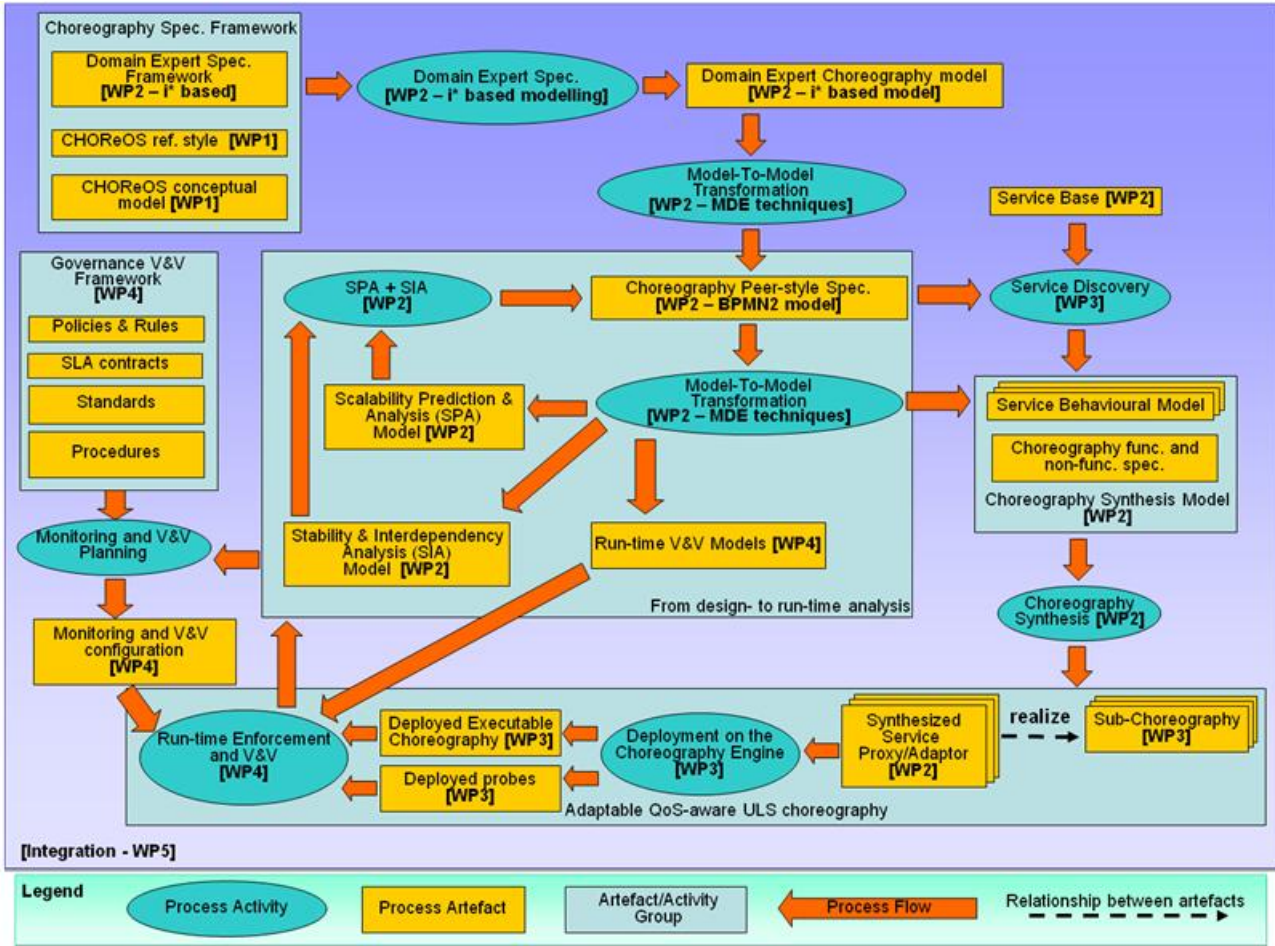


Figure 12: The CHOReOS Development Process Model

- The comprehensive and perpetual *process extends to run-time* and does support domain expert developers and (end-)users in all the phases of the choreography life cycle. Such a process has to explicitly account for *complex analysis and validation steps at run-time* when all the necessary pieces of information are available. In other words, some activities of the process may be executed at run-time, as specific internal capabilities of the choreography engineered through the process itself. Furthermore, the process unifies the developer and user roles by considering the user as taking an active part in the process that controls and affects the choreography development and its overall behavior.

In the following, we present the CHOReOS initial conceptual model by separating functionality, structure, and behavior views. Figure 13 shows an overview of the CHOReOS concerns by highlighting the NEXOF-RA ones that have been extended (see the label *extended*), and the ones that have been left as they are at the current stage of the project (see the label *as it is*).

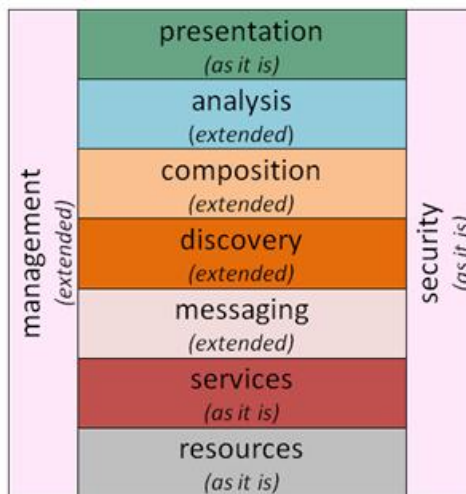


Figure 13: Current situation of the CHOReOS Concerns with respect to the NEXOF-RA Concerns

4.1. Composition Concern

Within the *Composition* concern (see Section 3.2), NEXOF-RA considers the concepts of *Orchestration* and *Choreography* as simple specializations of a service composition. Despite the emerging importance of the concept of choreography in the context of the Future Internet, NEXOF-RA just mentions choreography, rather than providing a clear and detailed definition.

As motivated in [D1.1], in the last few years, building upon the wide acceptance of the service-oriented architecture paradigm, there is a growing interest in choreography as a *key concern* in forming complex service-oriented systems. Choreography is put forward as a generic abstraction of any possible *collaboration* among multiple services, and integrates previously established views on service composition, among which service orchestration. In principle, any possible view on collaboration and previously established view on composition among multiple services (e.g., service orchestration) can be abstracted as a choreography. Three often overlapping viewpoints and related terminologies can be distinguished [DD04, BDO06]: (1) *Choreography* captures collaborative processes involving multiple services and their interactions from a global perspective; (2) *Behavioral interface* (see *Service Behavioral Model* in Figure 12) captures the behavior of a single service that participates in the choreography (i.e., the signature and the interaction protocol with the environment that the service supports); and (3) *Orchestration* deals with the description of the interactions in which a set of services can be engaged, as well as the internal steps between these interactions.

The above considerations have led us to extend the *Composition* concern of NEXOF-RA to address the peculiarities of choreographies and their distinctive features that are not currently covered by NEXOF-RA (see Figure 14 that shows only the new/refined functionalities). According to Figure 14, a **Choreography Designer** should have the possibility to **specify** a service choreography with respect to the **Goal and Requirement Specifications** provided by the **Domain Expert** and **Consumers**. The **Creation of a Service Choreography** encompasses its **specification**, and its **implementation**. The implementation of a service choreography includes the **synthesis** of proxies/adaptors for suitably coordinating the involved services. Finally, the **Operator** has the responsibility of enabling the **execution** of the created choreography, which can then be **enacted** by the consumer as needed.

In the sequel, the proposed extensions will be detailed by showing the structure of the provided extensions of the composition concern (Section 4.1.1), and the data flows of the proposed functionalities (Section 4.1.2).

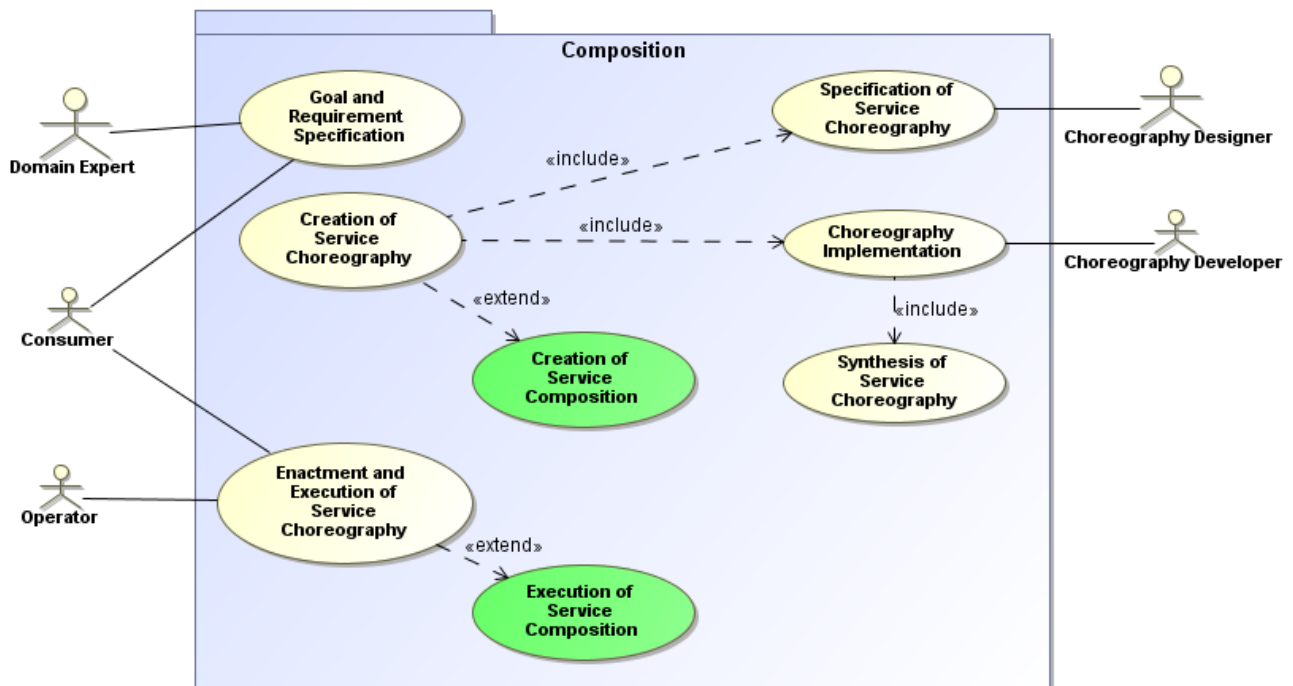


Figure 14: New functionalities in the refined Composition Concern

4.1.1. Structure View

Figure 15 shows the structure view of the composition concern showing only the extensions related to the **Choreography** concept. A **Choreography**, as described by a **Choreography Model** (specializing the **Composition Model**), is related to (i) the ability to model business contracts, i.e., **Conversations** in terms of a set of **Messages** which can be **exchanged** (possibly according to a **Message Exchange Pattern**) among multiple **Participants** and (ii) how a business contract is “enforced” with respect to the individual **Behavioral Model** of participants. The **Behavioral Model** defines the interaction behavior in terms of allowed sequence of **Messages**.

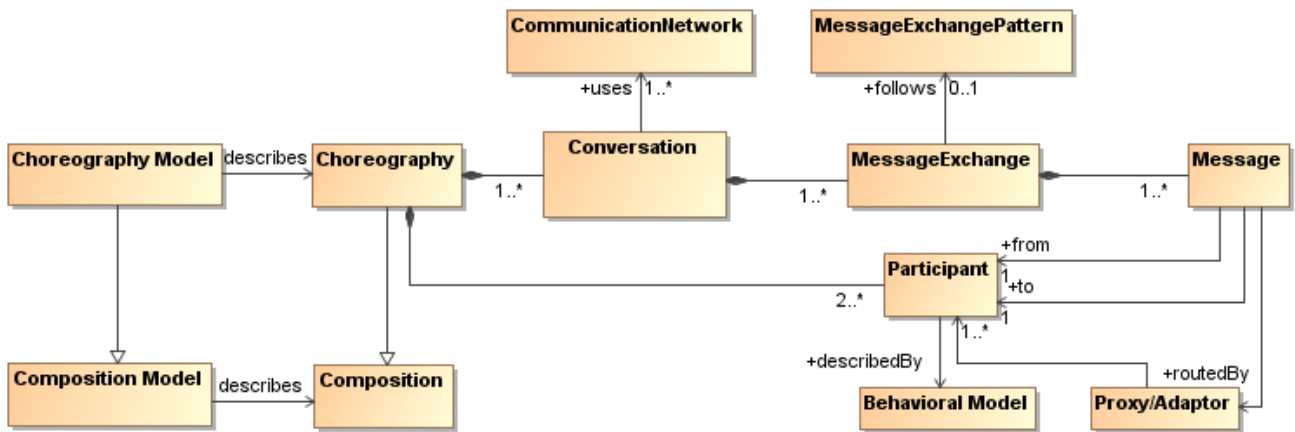


Figure 15: Structure View of the Composition Concern

To summarize, a choreography specifies how messages are exchanged among participants in a global conversation. Thus, the focus is not on orchestrating the work performed within each participant, but rather on the external exchange of information (messages) between these participants.

4.1.2. Data Flow

Building upon the previous discussion, a choreography can be seen as a collaborative process among interacting participants. The message exchange globally defined by the choreography has to be projected among different participants according to the “local” roles (in a peer-style fashion) they play to fulfill the global choreography. In the following, the functionalities **Specification of Service Choreography**, **Goal and Requirement Specification**, **Synthesis of Service Choreography**, and **Enactment and Execution of Service Choreography** are described. The **Discovery of Service** is described in Section 4.4.

Specification of Service Choreography

According to Figure 16, the **Specification of Service Choreography** takes as input **Choreography Patterns** as derived by the specification of the choreography goal and requirements defined by domain experts and refined by consumers (see Figure 17). Then, the service choreography specification process produces a **Choreography Model** (named domain expert choreography model in Figure 12). Such a model is taken as input by a service choreography **synthesizer** together with **Service Descriptions**. The latter describes the services that can be potentially used to realize the choreography. These services have been discovered from the **Service Base(s)** (see Section 4.4) with respect to the specification given in the **Choreography Model**. As further detailed in the following, the output of the service choreography synthesis is a set of **Proxies/Adaptors** which are generated to distributedly support **choreography enactment**, and hence for enabling the choreography realization. In the overall process, MDE techniques in terms of model transformations and generative techniques can help the management of some FI characteristics like its ultra-large scale. Moreover, to make the synthesis approach scalable and, hence, to address the large scale size of the FI compositional reasoning techniques can be adopted for reducing the state explosion problem which typically affects synthesis processes.

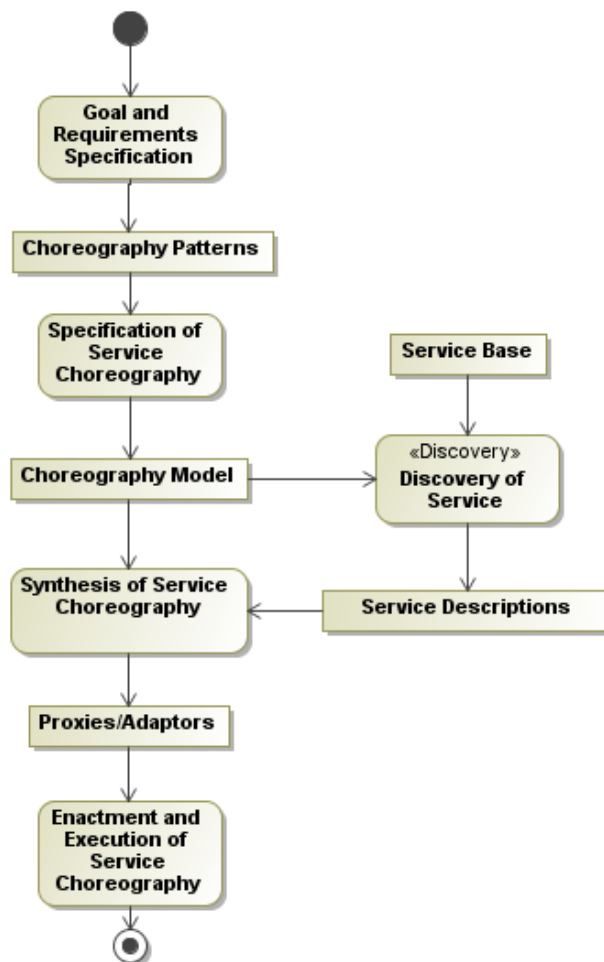


Figure 16: Data Flow in the Composition Concern

Goal and Requirements Specification

Requirements are expressed by **Domain Experts** (domain specific requirements) and **Consumers** (user needs) using a structured approach that includes qualitative qualifiers, satisfaction ratings and prioritization (see Figure 17). The domain expert oversees the process, receiving the consumer requirements, either directly or indirectly through the **Evolving Requirements Specification**. The resulting set of prioritized quality-based requirements is then associated with choreography strategies, which are expressed in the forms of **patterns**.

The service choreography patterns encapsulate different complex choreography decisions made in the presence of user requirements. Each provides a structure for associating user requirements with

different possible choreographies of classes of services (i.e., groups of similar services see Section Discovery Concern4.4) that, when implemented, deliver service-oriented applications with different qualities. Each pattern considers the qualities of the services to be invoked, the qualities that different choreographies offer and how, when combined, these service and choreography qualities deliver systems of different qualities. Users of CHOReOS solutions will match their user requirements to the service choreography patterns that will inform more detailed choreography design.

The goal and requirement specification techniques will support scalability, as testified by the adoption of *i** [YMY94,FG09], which has previously been applied to model large-scale systems [MLR07]. As discussed in Section 5, we can draw on this modeling experience to ensure scalability of CHOReOS solutions.

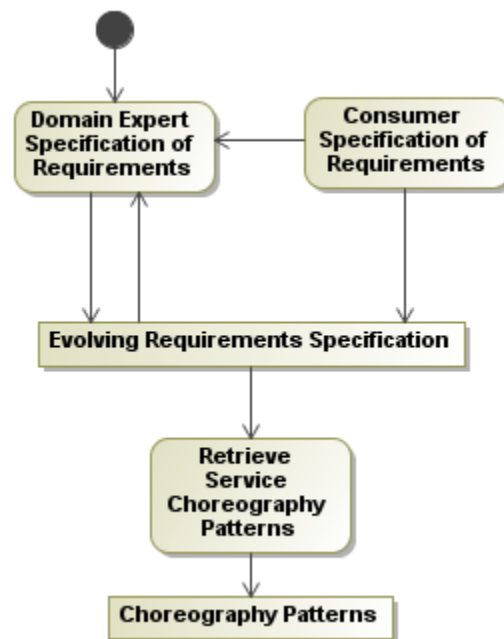


Figure 17: Functionality - Goal and Requirements Specification of the Composition Concern

Synthesis of Service Choreography

The synthesis of service choreography concerns the realization of distributed proxies/adaptors that, supported by the CHOReOS middleware, cooperatively work to support the enactment of the choreography.

By relying on suitable mechanisms such as the ones proposed by INRIA and UOI to organize hierarchically the vast amount of available services [AZI09b] within the service base, and by considering the coordinator synthesis approach described in [TII08] as baseline, **Model-to-Model Transformation** techniques [MHZ06, HMY06] will be introduced to refine the **Choreography Model** together with the service **Behavioral Models** into a **Peer-style Specification** (see Figure 18). Service **Behavioral Models** are obtained by the **Synthesis of Behavior Protocol** activity starting from the **descriptions** of services that have been discovered within the **Service Base** (see Section 4.4). We will build on the approach described in [BIPT09] for deriving such behavioral models.

The synthesis process must be able to understand how to actually coordinate the discovered services to suitably realize the specified choreography. For instance, it might be the case that the discovered services, although potentially suitable in isolation, when interacting together can lead to, e.g., concurrency and interaction mismatches such as deadlocks, or safety and timeliness violations. Thus, although a given set of services, if coordinated in the right way, can be used to achieve the specified choreography, they can completely miss the choreography's goal if coordinated in a different way.

Further applying **Model-to-Model Transformation**, the synthesis method will produce (possibly with partial human intervention/inspection) abstract **proxy/adaptor models**. Such models, accounting for service functional and non-functional abstractions, force the collaboration of the discovered services to guarantee the specified choreography. To actually realize the choreography, the abstract proxy/adaptor models are then concretized into actual software artifacts by means of **Model-to-Code Transformations**. The generated proxies/adaptors are deployed (see Figure 12 and Figure 19) on

the CHOReOS choreography engine and, by means of the CHOReOS service-oriented middleware, they suitably access and coordinate the discovered services.

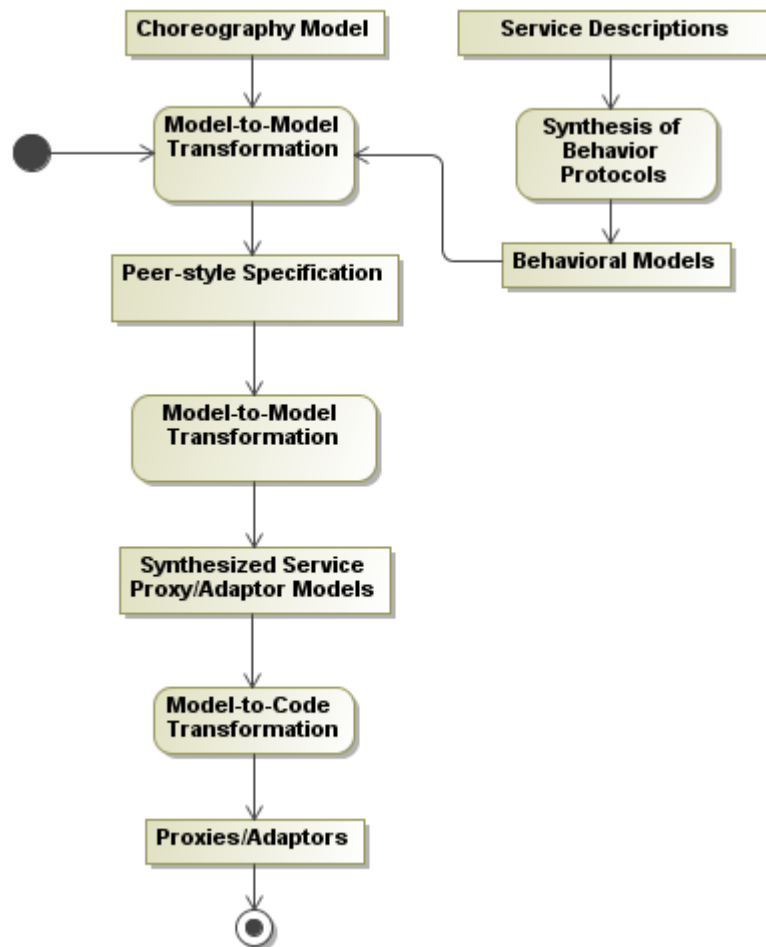


Figure 18: Functionality - Synthesis of the Composition Concern

Enactment and Execution of Service Choreography

Services **Proxies/Adaptors** are **deployed** in the execution environment providing services (see Figure 19). Meanwhile, a **Subscription** activity is triggered in order to set **Probes** on the **deployed services**. Probes enable the monitoring of running services and the control of their runtime performances. **Runtime V&V Model** results from the analysis activity that is realized **From Design Time to Runtime**. These models can then be considered in order to operate **Runtime Enforcement and V&V governance** on services and probes. In order to enable services collaborations, governance and V&V rules need to be respected.

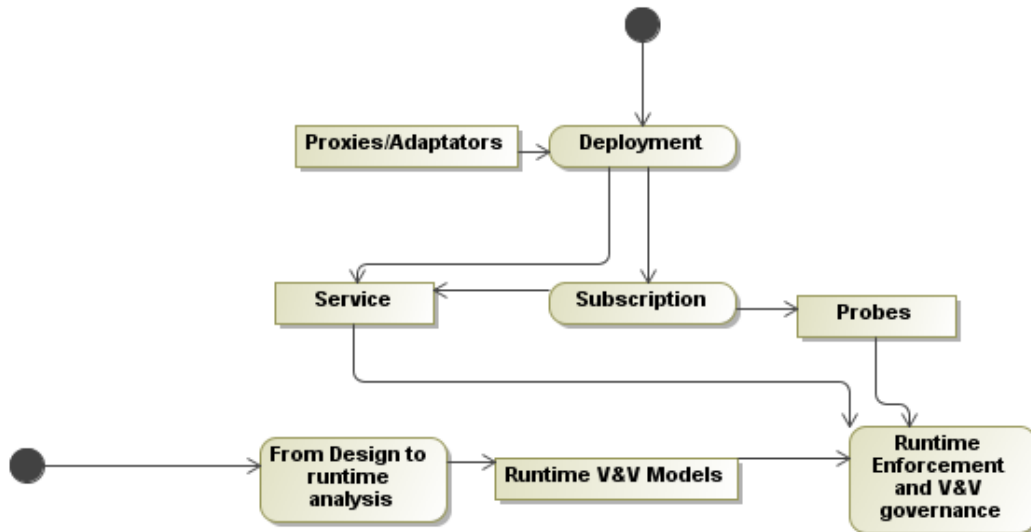


Figure 19: Functionality - Enactment of the Composition Concern

4.2. Presentation Concern

The Presentation concern of NEXOF-RA addresses all mechanisms to enable human users to interact and make use of the functionalities provided by the overall service infrastructure. It aims at supporting developers to work in a highly personalized environment to assemble composite end-user applications from a set of uniform building blocks. This concern strictly relates to the CHOReOS IDRE that will be developed in WP5 whose requirements are elicited in [D5.1]. We then intend to extend/refine the presentation concern with respect to the experiences gained in WP5.

4.3. Analysis Concern

The Analysis concern has been considered by NEXOF-RA as one of the important aspects that need to be investigated to derive a complete classification of *SOA for Future Internet* concepts. Nevertheless, due to the prioritization of research activities in the NEXOF-RA project, the concern has not been developed further. Such an aspect will be, on the other hand, deeply investigated within CHOReOS, in particular as a result of the activities that will be carried on within WP4, which is in large part devoted to Verification & Validation (V&V) aspects.

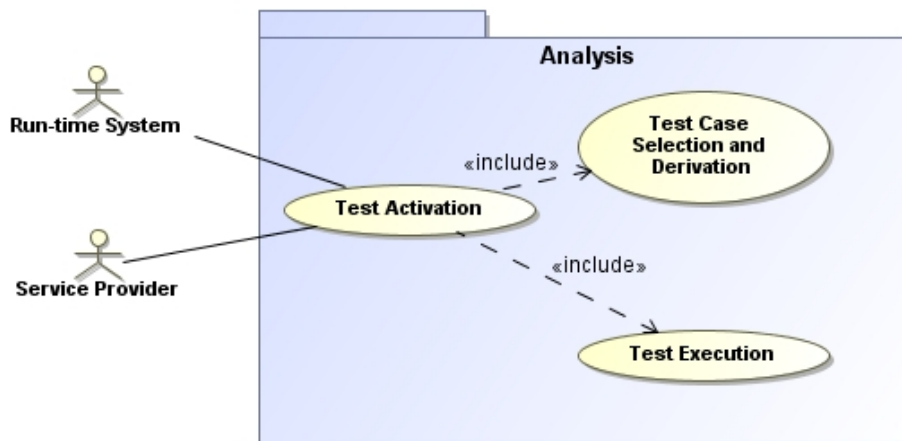


Figure 20: Functionalities of the Analysis concern

The Analysis concern will be strongly related to most of the other concerns and will have to consider and suggest possible integration to them. In particular, the Analysis concern can be immediately related to:

1. the extended *Composition concern* whose concepts will be used to identify strategies and techniques to apply for verification purpose;
2. the *Service concern* in order to identify the information describing a service and to define how services can be accessed and integrated;

3. the *Security concern* is also related to the analysis concern since activities to be carried out for analysis purpose could depend also from security settings;
4. the *Messaging concern* can suggest techniques to be applied for automatic test case derivation and analysis.

In addition, analysis activities will be strongly related and enabled/disabled in relation to governance setting and activities. The interrelation will be exploited in detail directly within WP4.

In CHOReOS, V&V activities will be activated both at design and run-time by subsystems that will be specifically developed for the purpose, as depicted in Figure 20. Testing activities can be activated both by the **Run-time System** environment or by a **Service Provider** to test specific services. The automatic activation tries to provide a solution to the possibility that the run-time context may change. V&V activities can then provide a useful mean to assess if the change could lead to possible failures. As detailed later, **Test Activation** includes **Test Case Selection and Derivation** and **Test Execution**.

4.3.1. Structure View

The CHOReOS partners will explore the possibility of extending test execution activities to the run-time phase. In order to make this possible, the CHOReOS infrastructure needs to be defined considering the concepts represented in Figure 21. In particular, the following elements are especially relevant:

1. **Test Selection Strategies**: this concept describes the algorithms that will be defined in order to derive test cases from choreography specifications, i.e., **Choreography Model**, and from information coming from the run-time environment (**Run-time Context**). This will make testing activities at run-time particularly effective and complementary to off-line testing activities, which cannot take into account real execution environment and parameter in the test derivation related activities.
2. **Test Agent**: represents those elements that permit to execute the defined test cases on real services. The verification will be mainly devoted with respect to choreography specification.
3. **V&V Manager**: includes the policy that underneath the decision of starting a test execution phase or other V&V activities. The V&V Manager will take its decision on the basis of **V&V events** coming from the run-time environment and distributed by an **Event Broker**, possibly implemented using Publish/Subscribe mechanisms.
4. **Test**, **Test Suite**, and **Test Case** describe the structure and the composition of test cases to be executed to evaluate a service.

Due to scalability issue, most of the elements in the structural models are available in more than one instance at run-time. As a consequence, the different elements will be dispersed within a CHOReOS infrastructure and will be able to act independently.

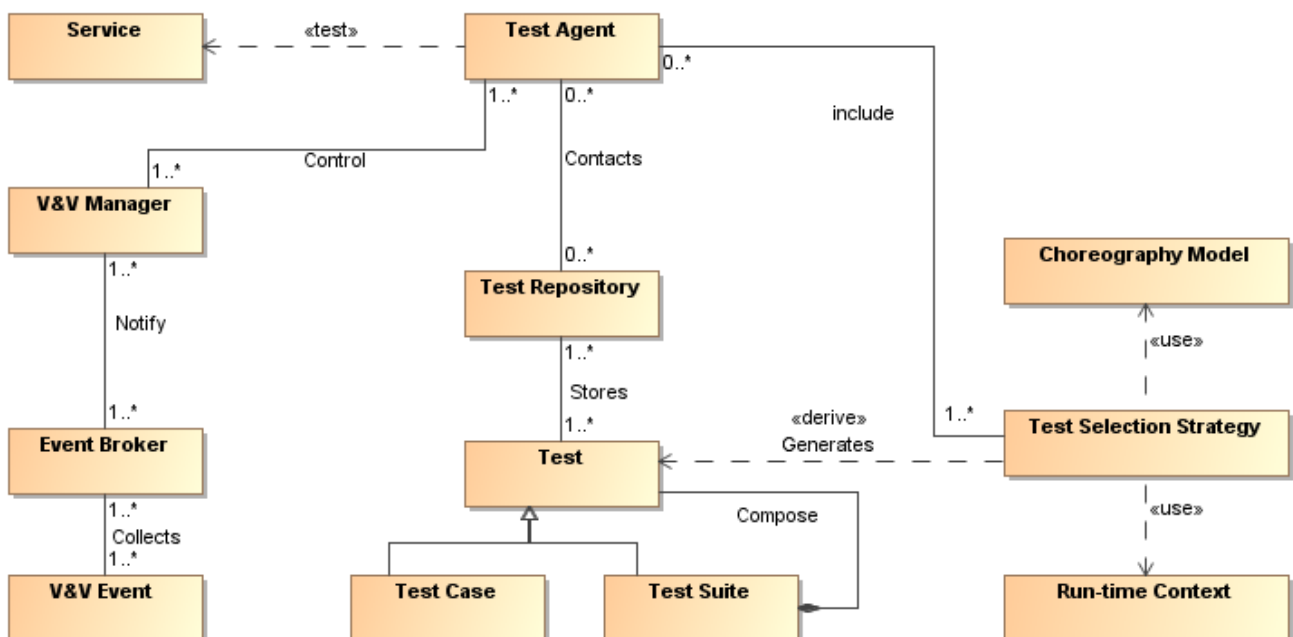


Figure 21: Structure View of the Analysis Concern

4.3.2. Data Flow

The run-time testing system will be mainly **activated** at run-time in order to execute a testing session to **evaluate** a service. With reference to the data flow diagram depicted in Figure 22, the activation is triggered by a specific **event** whose occurrence leads to verification activities. Such event could be part of management activities. After the activation, a **Test Case Selection** activity is conducted and this will result in test and test suites generation. Finally, generated tests are **executed** against the service under test. At this stage, we do not explore the possibility of performing specific activities in the case where executed test cases have highlighted behaviors that are not in-line with those specified in the considered choreography.

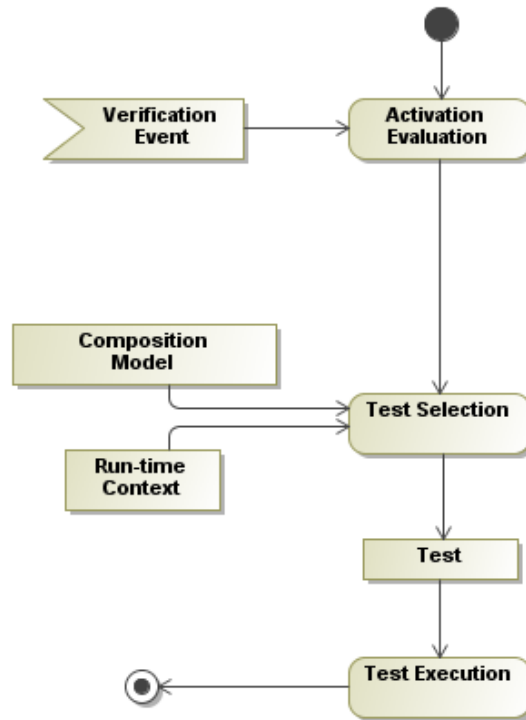


Figure 22: Functionality - Test Activation of the Analysis Concern

4.4. Discovery Concern

Typically, service discovery is based on a particular protocol that dictates how to publish and discover information about available services. Publication and discovery further involve the use of service registries (which we also call *bases*) that store information about available services. As detailed in the discovery concern of NEXOF-RA, the main actors that relate to the discovery concern are **Providers** of services and **Consumers** of services (see Figure 23). Moreover, we may have service **Brokers** that act as mediators. The typical functionalities of a service discovery protocol are: (1) the **Publication of Service Offering**, which is used by the service providers; (2) the **Discovery of Services**, used by service consumers; and (3) the **Publication of Service demands**, which is also used by service consumers. The discovery functionality may be decomposed in two alternative functionalities; the first one allows service consumers to **browse** the contents of registries, while the second functionality allows **specifying a service demand** and automatically matching this demand against the descriptions of available services towards finding the ones that satisfy the issued demand.

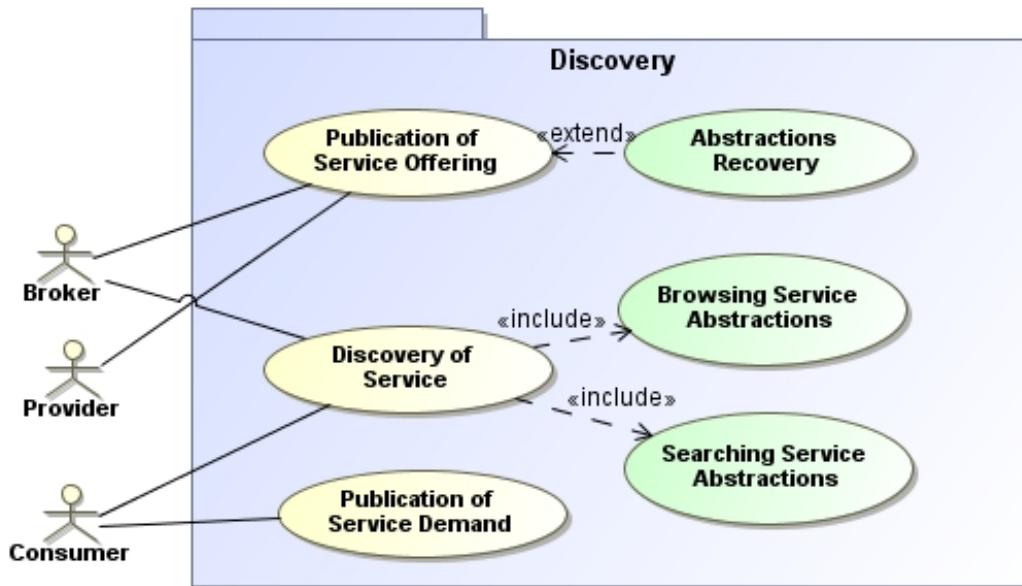


Figure 23: Extended Functionality of the Discovery Concern.

Concerning the service registries that support the aforementioned functionalities, a main concept highlighted in NEXOF-RA is the structure of their contents, which is typically based on a catalogue schema that divides services in different categories. Then, *a main issue to be handled in the context of the Future Internet is that this schema cannot be static or predefined*. On the contrary, it should be *scalable and reflect a proper categorization* of the heterogeneous functionalities of the vast amount of services that become available over time.

To this end, in CHOReOS, we go beyond NEXOF-RA by introducing new high-level concepts in the discovery concern, along with corresponding lower-level concepts in the structural and the data-flow views of this concern. Specifically, our contribution is summarized in the following points:

- A new high-level concept is introduced in the discovery concern, namely the **Abstractions Recovery** functionality. The purpose of this functionality is to reverse engineer **service abstractions**, i.e., groups of available services that are characterized by similar functional/non-functional properties (i.e., facets in NEXOF-RA terminology), out of services that become available over time. In a sense, the concept of service abstractions corresponds to categories. As detailed in the rest of this subsection, service abstractions are hierarchically structured.
- Moreover, the concept of service abstractions is employed so as to facilitate the CHOReOS browsing and searching functionalities of the service discovery concern. In particular, as showed in Figure 23, **browsing** and **searching** is based on descriptions of **service abstractions**, instead of descriptions of services, so as to enable the consumers to efficiently and effectively retrieve multiple candidates of services that offer similar properties.

4.4.1. Structure View

The **Abstractions Recovery** functionality, along with the concept of service abstractions are the central concepts that affect the definition of the CHOReOS information model of the service discovery concern.

- As showed in Figure 24, a **service abstraction** consists of a set of grouped **services**. Each service is associated with a corresponding **service description**.
- A service abstraction is further associated with an abstract service description. The abstract service description comprises a set of **facets** that represent the common/similar facets of the grouped services.
- A service abstraction may be associated with other lower-level service abstractions that represent groups of services, which offer more and possibly more concrete functionalities. Moreover, a service abstraction may be associated with higher-level service abstractions that represent groups of services, which offer fewer and possibly more abstract functionalities.

We distinguish the following different types of abstractions:

- **Functional abstractions** - The main purpose of a functional abstraction is to represent a group of services that offer similar functional properties (i.e., functional facets). At a glance, the description of a functional abstraction is characterized by a set of abstract functional facets, such that there is **mapping** between the abstract facets and the concrete functional facets of the services that are represented by the functional abstraction. To facilitate choreography adaptation and specifically the substitution of the services used in a choreography, the mapping must preserve certain **well-formedness rules** of behavioral subtyping, adapted to the specificities of service-orientation [AZI09b]. The definition of relationships between higher and lower level abstractions should also be based on such rules. A functional abstraction may be associated with non-functional abstractions, defined in the following.
- **Non-functional abstractions** - The main purpose of a non-functional abstraction is to represent a group of services that are characterized by similar non-functional property(ies) (i.e., non-functional facets) such as performance, reliability, availability, reputation, cost. Hence, the description of a non-functional abstraction is characterized by abstract non-functional facets that reflect the range of the concrete non-functional facets of the grouped services.

Based on the aforementioned functional and non-functional abstractions, the **catalogue schema** of a service registry (i.e., **Service Base** in CHOReOS terms) can be organized in terms of different **functional** and **non-functional views** that can be browsed or searched with respect to given service demands. The **results** of the browsing/searching are groups of services that come along with the service abstractions that represent them.

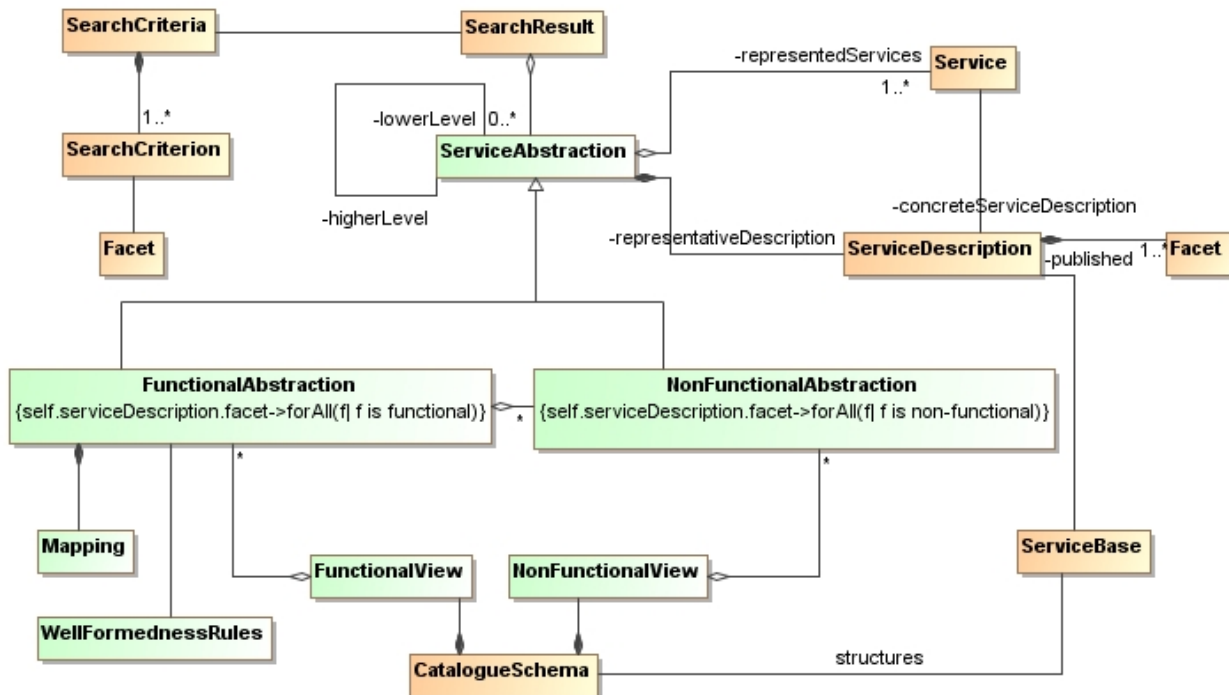


Figure 24: Structure View of the extended Discovery Concern.

4.4.2. Data Flow

Following, a new data flow is introduced in the service discovery concern to describe the abstraction recovery functionality. Moreover, the rest of the CHOReOS functionalities are defined according to the notion of service abstractions.

Abstractions Recovery

The abstraction recovery functionality may be periodically initiated or on-demand, depending on the availability of new services over time (Figure 25).

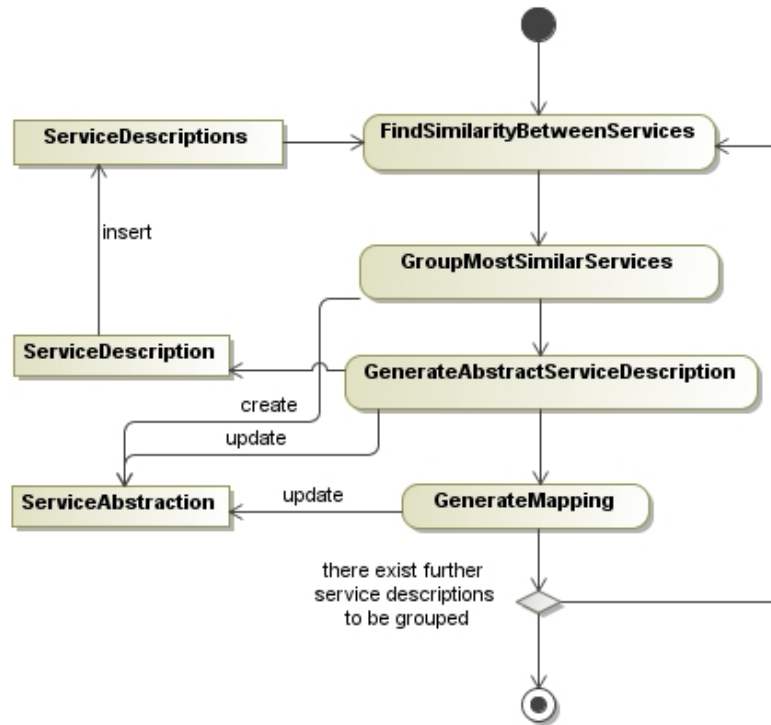


Figure 25: Functionality - Abstractions Recovery of the Discovery Concern

The input to the functionality is a set of available service descriptions. Then, the basic activities involved in the recovery of service abstractions include:

- The calculation of similarities between the functional/non-functional properties of the available services.
- **Finding** and **grouping** of the most similar service descriptions.
- The **generation** of an **abstract service description** for each particular group of similar services.
- The **generation of mappings** between the properties of the abstract service description and the properties of the grouped services.
- The whole process is iterative; the abstract service description of each **service abstraction** produced by the recovery process is inserted in the input set of available service descriptions, so as to enable the creation of higher-level abstractions in subsequent iterations of the recovery process.

The similarity criteria and the metrics used for the recovery of service abstractions depend on whether the goal is to produce functional or non-functional abstractions.

Publication of Service Offer

The publication of service offer functionality allows the advertisement of information about a service provided as input by a corresponding service provider (Figure 26).

The **subscription** activity may further trigger the **abstraction recovery** functionality depending on whether a set of published services can be categorized in already existing service abstractions or not. In the latter case, the **functional** and/or the **non-functional views** of the registry should be updated via the execution of the abstraction recovery functionality.

A service provider may further request to **change**, or **remove** a particular service offer that was previously published. These requests may trigger the need to **update** the contents of a service abstraction included in the functional and/or the non-functional views of the service base.

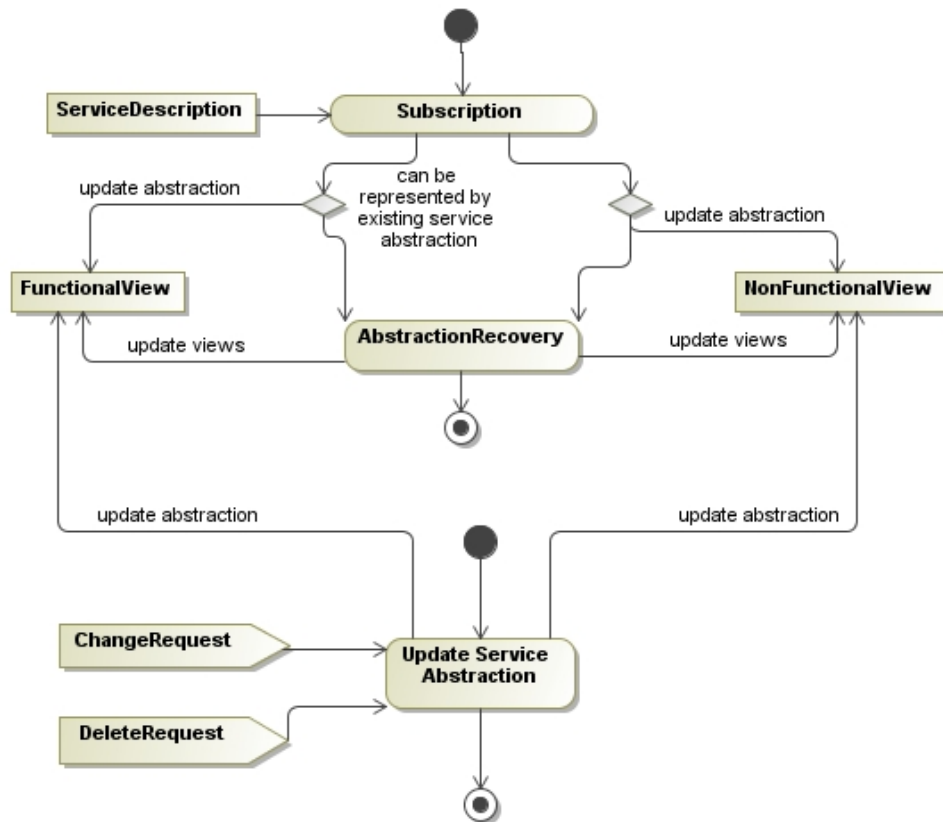


Figure 26: Functionality - Publication of Service Offer of the Discovery Concern

Discovery of Service

As specified in the NEXOF-RA discovery concern, the discovery of services can be done either by browsing the content of the registry or automatically via algorithms that match a given service demand against the service descriptions stored in the registry. As previously mentioned, the concept of service abstractions facilitates the efficient and effective retrieval of multiple candidates of services that offer similar properties (Figure 27). More specifically:

- According to the CHOReOS **browsing for abstractions** activity, the consumer explores the abstractions hierarchies of the **functional** and **non-functional views** of the registry to locate groups of services that offer similar properties, which satisfy the consumers **service demand**. By construction, the abstractions must be such that, if the properties of a service abstraction satisfy the demand of the consumer, then any service represented by this abstraction may also satisfy the demands.
- The CHOReOS **searching for abstractions** activity, realizes means that allow matching the consumers demands against abstract service descriptions that characterize abstractions, which group services with similar properties. As also mentioned in the case of the browsing activity, the abstractions must be constructed such that if the matching of a service demand with a service abstraction description is successful, then any of the services that are represented by the matched service abstraction may be a candidate solution that satisfies the issued service demand.
- Then, the outcome of both of the aforementioned activities is a set of services that meet the required service demand; the group of discovered services comes along with the abstraction that represents the discovered services.

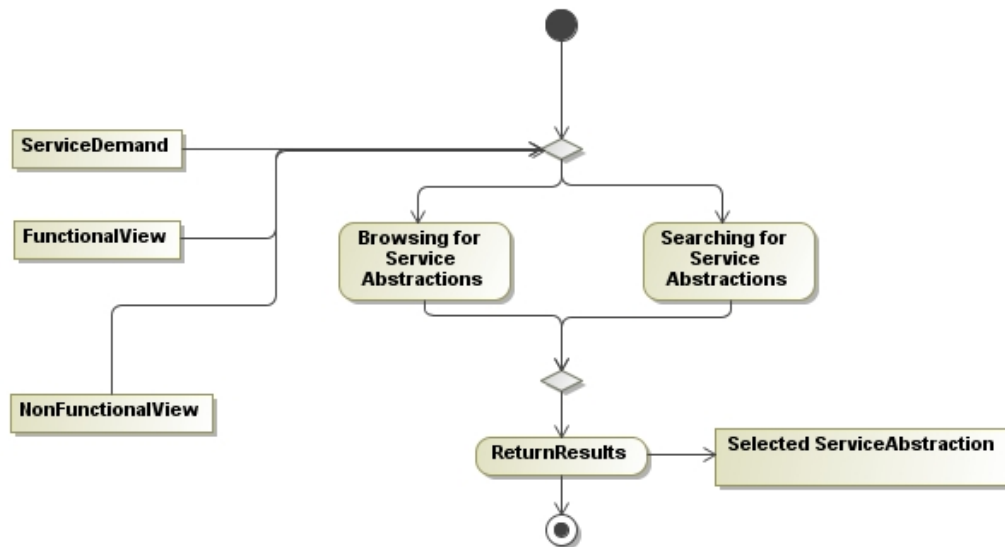


Figure 27: Functionality - Discovery of Service of the Discovery Concern

Publication of Service Demand

The publication of service demand is an asynchronous way for discovering services (Figure 28). Consumers register their interest for services along with a specific service demand specification. Then, at any time a new service offer is published, it is checked against the demands (based on the typical searching for service functionality) of the consumers and corresponding notifications are sent if the demands are satisfied by the new service offer. Obviously, the complexity of this particular functionality is proportional to the complexity of the typical searching for service functionality, due to the fact that the search functionality must be triggered for every consumer publication of service demand. The concept of service abstraction may be employed to reduce the aforementioned complexity. In particular we adapt the publication of service demand functionality as follows.

- According to the subscription activity, a consumer provides as input a **service demand**.
- Based on the service demand the **Searching For Service Abstraction** activity is triggered.
- If this activity results in a service abstraction that satisfies the service demand, then the **demand is associated** with this service abstraction.
- Subsequent publications of new service offers that result in new members of the service abstraction, trigger the **retrieval of associated consumers**.
- **Notifications are sent** to the associated consumers, without the need for matching the service demands against the service offers.

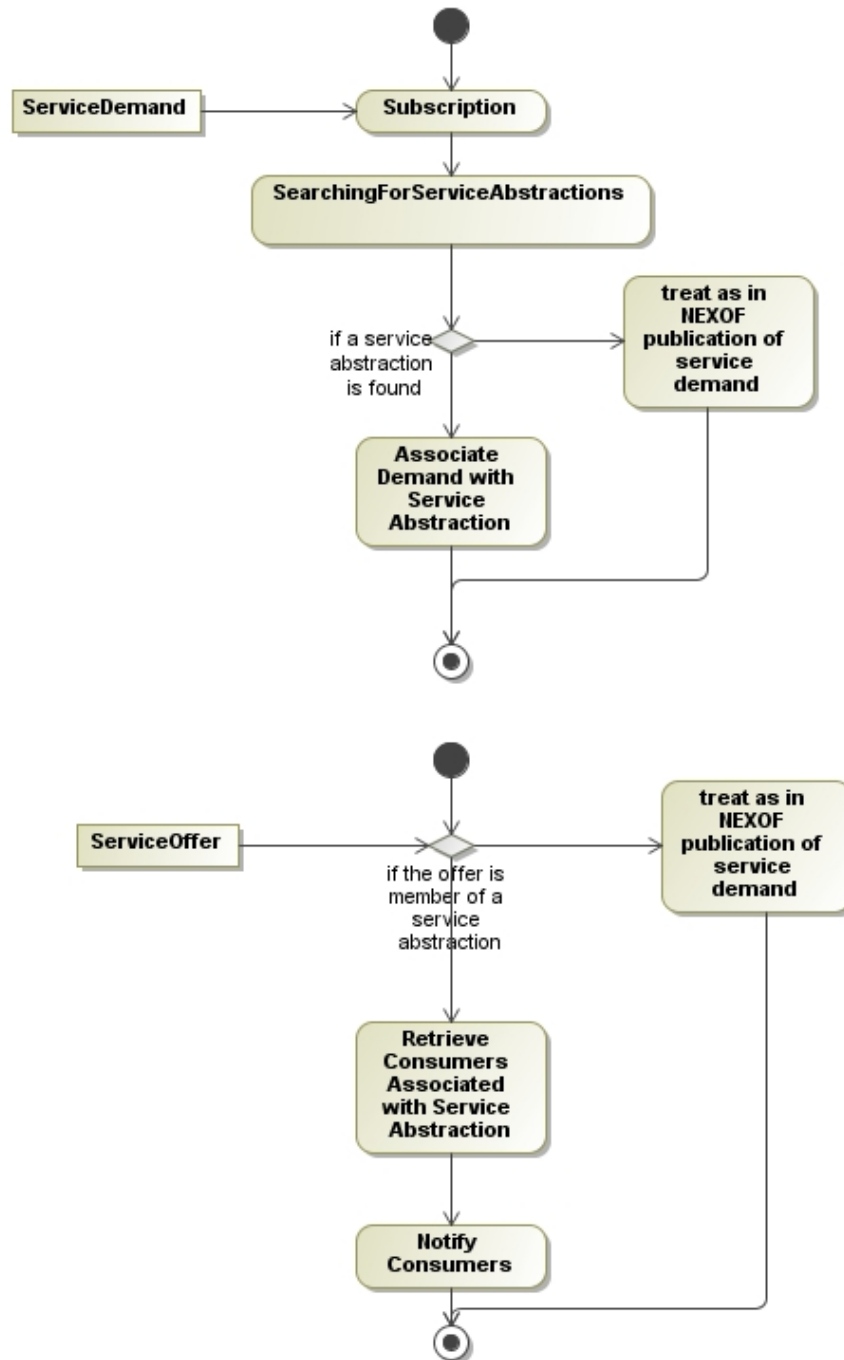


Figure 28: Functionality - Publication of Service Demand of the Discovery Concern

4.5. Message Concern

The **Message concern** of the NEXOF-RA conceptual model [NEXOFRM10] addresses in a generic way the communication capability that allows applications or services to interact with other services. This interaction is assumed to be performed through the exchange of messages. The generality of the approach makes the NEXOF-RA message concern directly applicable to the initial CHOReOS conceptual model. Nevertheless, the CHOReOS vision of the Future Internet calls for an extended viewpoint in the way this message concern is perceived within the CHOReOS conceptual model.

As stated in our literature survey and CHOReOS definition of the Future Internet in Chapter 2, the Future Internet vision aims at incorporating all current networking environments into a single, ubiquitous setting connecting services, people and things. This results in high diversity in employed communication middleware and calls for support for the heterogeneous coordination/interaction models, such as remote procedure call (RPC), message-based, shared memory, and event-based

models, employed in the various middleware [GRAPI11]. Indeed, different interaction models apply to different needs; for instance, asynchronous, event-based publish/subscribe is more appropriate for highly dynamic environments with frequent disconnections of involved entities. This fact makes the various service bindings accounted for in current service-oriented computing too stringent, since they comply to a single – i.e., RPC or message-based – interaction model. Hence, the CHOReOS choice of choreography as global coordination style among services should further be underpinned by support for the heterogeneous interaction models.

Furthermore, since entities interacting in ad hoc settings cannot be assumed to share the same interaction model, interoperability among the models should be enabled. Despite a number of approaches dealing with interoperability between interaction models, provided solutions are in general ad hoc and concern specific cases. In the Future Internet as perceived by CHOReOS, an overall solution to this issue is required, based on appropriate modeling abstractions and transformation mappings between models. Moreover, a precise evaluation of such mappings with respect to the preservation of semantics is needed. The first step towards a systematic solution to the above issue is to incorporate adequate representation in the CHOReOS Conceptual Model. Hence, the message concern of the CHOReOS Conceptual Model should reflect the Future Internet coordination/interaction diversity and interoperability. In the following, we reconsider the NEXOF-RA message concern from the CHOReOS Conceptual Model point of view.

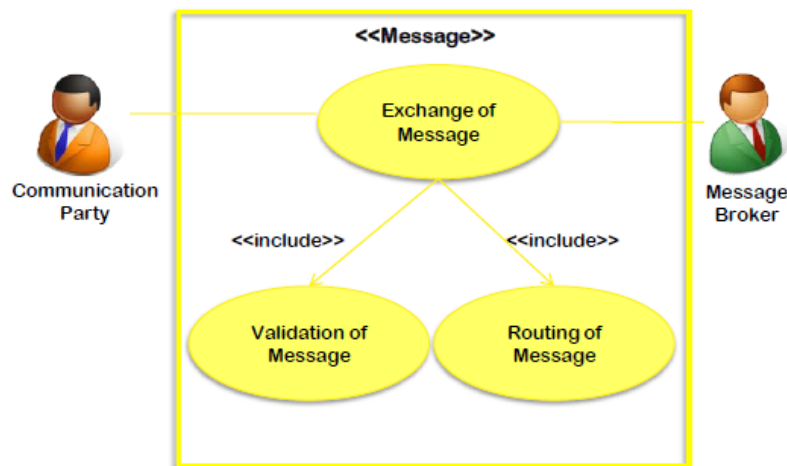


Figure 29: Functionalities of the Message Concern [NEXOFRM10]

The functionalities of the message concern are depicted in the use case diagram of Figure 29, borrowed from [NEXOFRM10]. As already stated, the principal functionality concerns the **Exchange of Message**, which is performed between **Communication Parties**, with eventual support from a **Message Broker**. The message broker provides functions that may range from simple message forwarding to message mediation (e.g., in terms of protocols and data formats) among **Communication Parties**, and can be, e.g., an ESB. In the CHOReOS conceptual model, we extend the message exchange notion to denote an end-to-end exchange of data (including control) between communication parties, where the semantics of this exchange may take different forms, such as synchronous or asynchronous, one-to-one or one-to-many, via a direct message or a brokered event or a write/read operation in a shared memory. As further depicted in the diagram, the exchange of message functionality includes the **Validation of Message** and **Routing of Message** functionalities, commonly supported by communication middleware. In the following sections, we discuss in more detail the structure view and the data flow view of the message concern.

4.5.1. Structure View

Figure 19, borrowed from [NEXOFRM10], depicts the information model of the message concern. The core concept is **Message**, basic unit of communication between parties. This concept remains perfectly valid in the CHOReOS conceptual model, while keeping in mind the extension given to the exchange of messages in the previous section. A message is characterized by a **Message Type**, classifying the structure and semantics of the message, and **Message Metadata** associated with the content of the message and used to process and deliver the message.

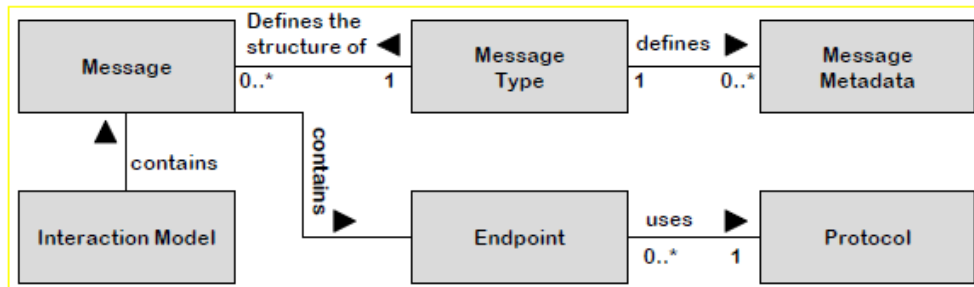


Figure 30: Structure View of the Message Concern [NEXOFRM10]

A message is further associated with an **Endpoint**, a transport **Protocol** and an **Interaction Model**. These concepts specify the entry point of the recipient service that this message should be sent to and also the interaction pattern and data transfer protocol that should be followed for correctly interacting with the service. In the NEXOF-RA message concern specification, it is stated that the interaction model may be synchronous or asynchronous and that it relies on a set of message exchange patterns (MEPs), where an explicit pointer is given to the MEPs defined in the SOAP specification, such as one-way and request/response. In the CHOReOS conceptual model, we extend the notion of endpoint to denote, besides a unique well-identified recipient service²⁶, a more loosely coupled destination, such as all the communication parties subscribed to a specific topic in a publish/subscribe setting, or all the parties reading a specific tuple from a shared tuple space. Then, the interaction model and transport protocol are extended accordingly to account for the diverse interaction models and associated transport protocols included in the scope of CHOReOS. Such interaction models and related MEPs include remote procedure call (RPC), message-based, shared memory, and event-based models.

4.5.2. Data Flow

In the NEXOF-RA message concern specification, a number of activity diagrams are introduced for defining the data flow of the message concern functionalities, as depicted herein in Figure 31, Figure 32, Figure 33, and Figure 34.

Figure 31 shows the top-level activity diagram for the message concern, where the functionality exchange of message processes an arriving message so that it is **dispatched** to the appropriate recipient party, while taking into account the input of **Validation Properties** that serve to validate the message. All this is perfectly applicable to the CHOReOS Conceptual Model, while dispatching now conforms to the extended message exchange meaning introduced in the previous sections.

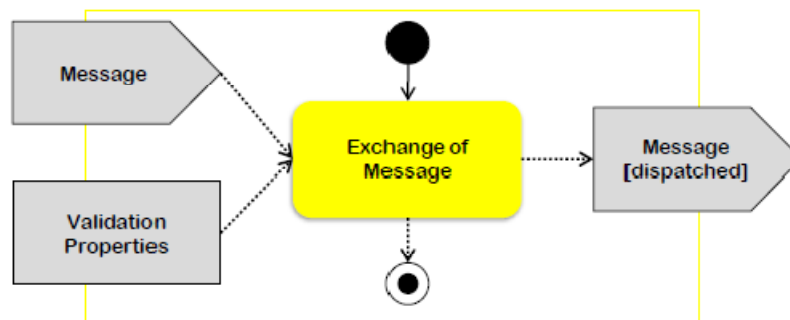


Figure 31: Data Flow of the Message Concern [NEXOFRM10]

The functionality **Exchange of Message** is analyzed in Figure 32 into the two sub-functionalities already mentioned, i.e., validation and routing. Validation checks the message based on the validation properties (e.g., with respect to its message type or for well-formedness) and validates it or issues a **Reject Notification** to be delivered to the sender of the message. A validated message is subsequently passed to routing.

²⁶ The case of message dispatching to one-of-many recipients and related load balancing supported by the message broker is also provided for by NEXOF.

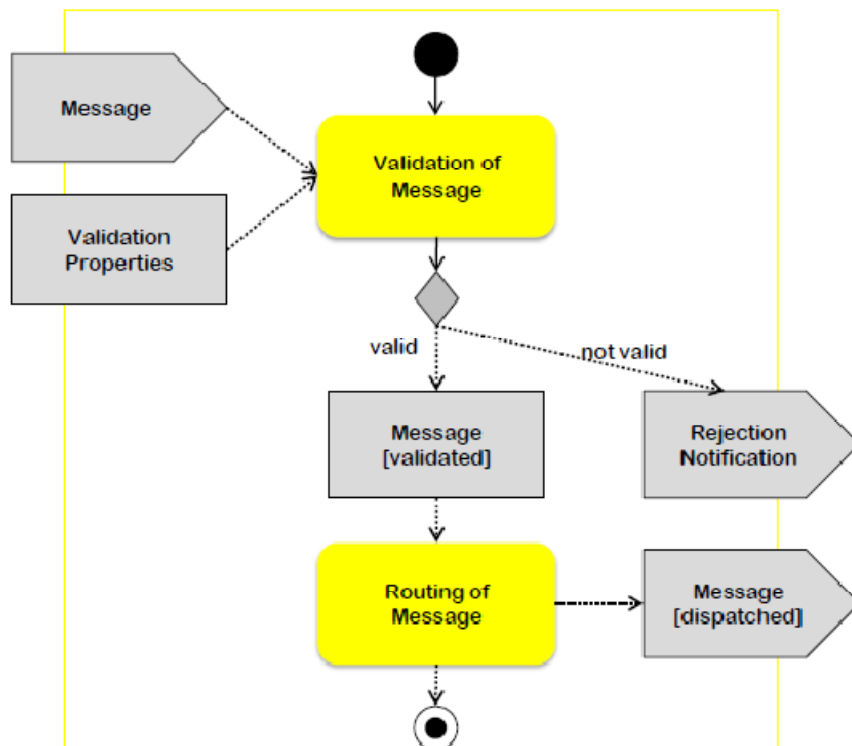


Figure 32: Functionality - Exchange of the Message Concern [NEXOFRM10]

The functionality **Routing of Message** is analyzed in Figure 33. A message is routed to its recipients according to their endpoint information. The message is first **forwarded** to the message broker, which takes care of possibly **adapting** the message and finally dispatching it.

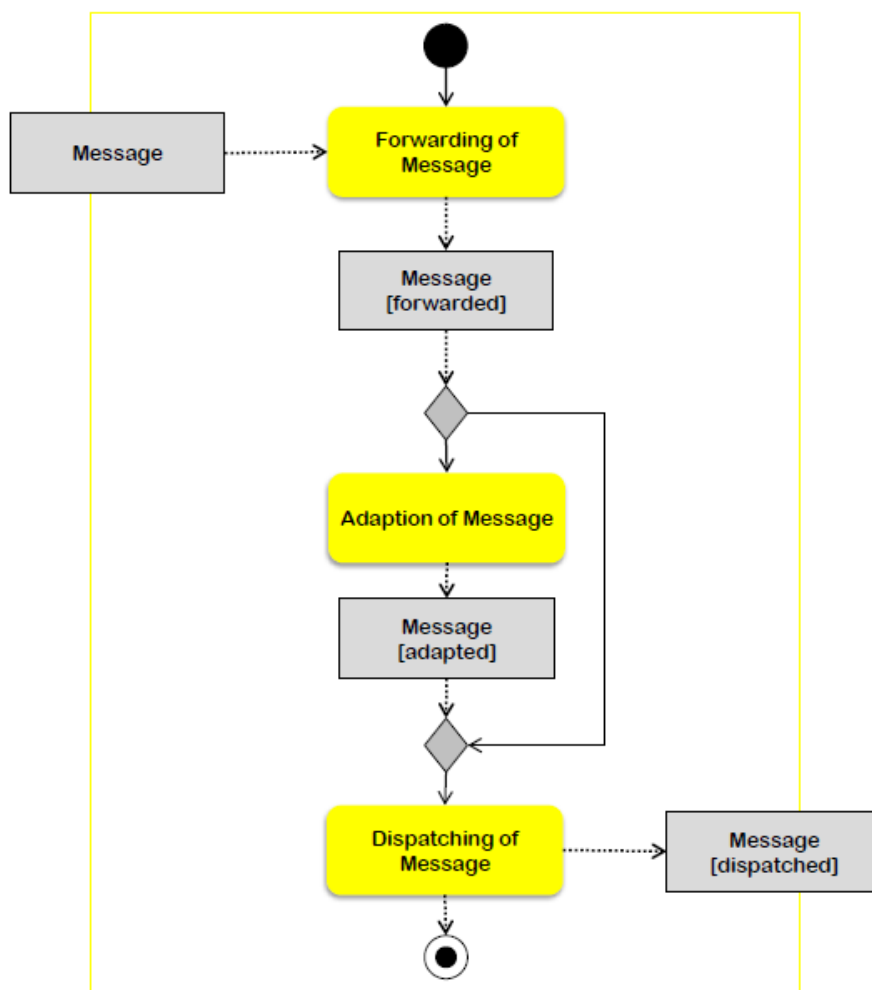


Figure 33: Functionality - Routing of the Message Concern [NEXOFRM10]

The functionality **adaptation of message** is analyzed in Figure 34. After having corrected some inconsistencies in the NEXOF-RA diagram (unnecessary message forwarding and duplication of the message input inside the adaptation functionality), we extend it with some additional functionalities, indicated in the figure with a different color and script code.

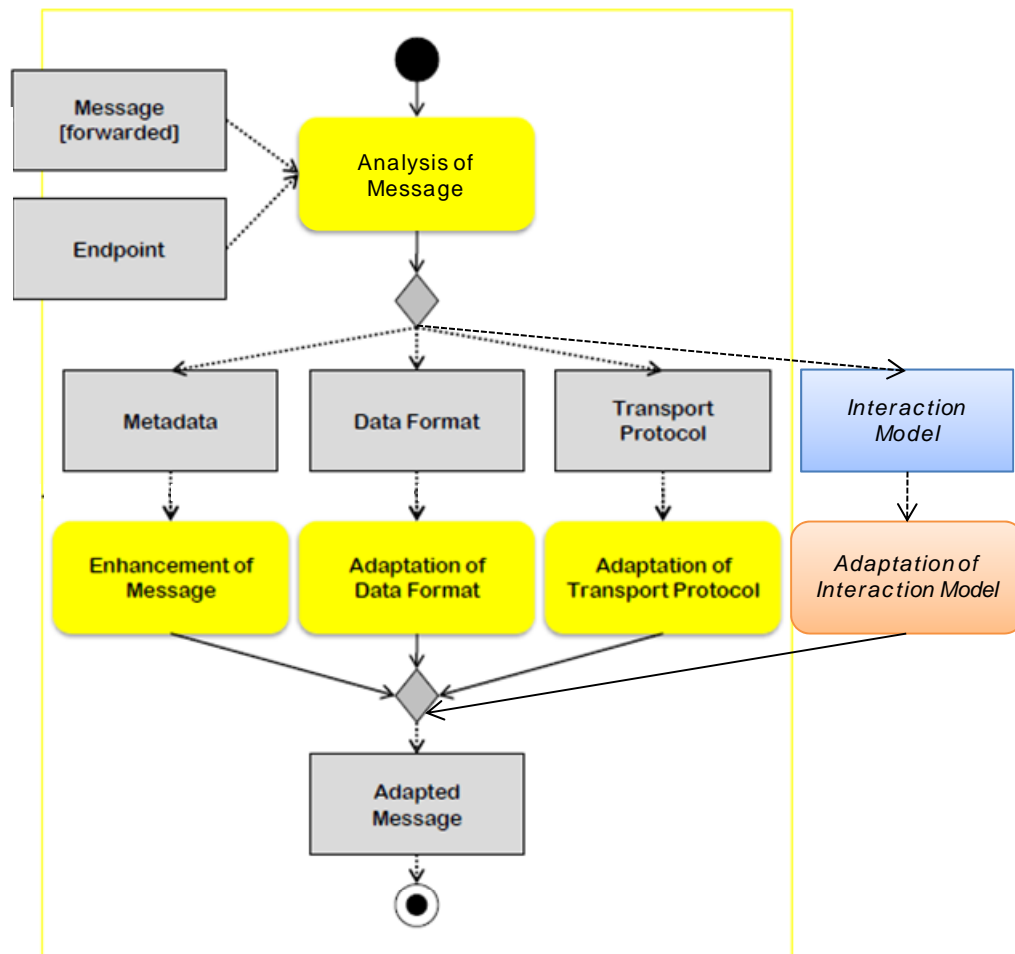


Figure 34: Extended Functionality - Adaptation of the Message Concern
(update of diagram from [NEXOFRM10])

Based on endpoint (i.e., recipient party) information, message adaptation may comprise different kinds of adaptation, such as **metadata adaptation** (or **enhancement of message** in the figure), **data format adaptation**, and **transport protocol adaptation** between the sender and recipient parties, when this is required due to related heterogeneity identified between them. In the CHOREOS Conceptual Model, we additionally deal with **Interaction Model** adaptation, the necessity of which has already been made clear from our discussion above.

A key issue pointed out by NEXOF-RA is that message adaptation must preserve the message semantics. Nevertheless, interaction model adaptation is more radical and may introduce some loss or modification of semantics. We extend/relax this condition in the CHOREOS conceptual model, by stating that transformation mappings between interaction models should preserve the semantics as much as possible. As we indicated already, we intend to include in our adaptation solution a precise evaluation with respect to the preservation of semantics.

4.6. Service Concern

The Service concern of NEXOF-RA is related to the design, implementation, and execution of service components. Moreover it concerns the promotion of legacy components to services. Even though important, these aspects are not in the main focus of CHOREOS, which instead focuses on the choreography of already available services. However, the definition of the service concern is called to evolve to reflect the heterogeneity of services in the Future Internet, and especially those originating from the Internet of Things. Related extension of the service concern will be defined in the next

version of the conceptual model, based on WP3 outcomes as well as definition of the CHOReOS architectural style to be investigated next within WP1.

4.7. Resource Concern

The resource concern of NEXOF-RA deals with the computational resources needed to support the execution of all the software components that constitute service platforms, e.g., storage, virtualization of computing and network resources. The generality of the approach makes the NEXOF-RA resource concern directly applicable to the initial CHOReOS conceptual model. Nevertheless, in the next version of the conceptual model, the concern will be refined in order to better fit the CHOReOS vision of the Future Internet, in particular accounting for resources associated with Cloud and Grid computing as well as with the Internet of Smart Things, as investigated within WP3.

4.8. Governance and Management Concern

In the NEXOF-RA conceptual model, a special interest is dedicated to the management concern at runtime. CHOReOS will extend this model to include governance aspects at both design- and runtime.

Governance can be defined by the trilogy of: doing the right things, the right way for the right stakeholders [Marks 2008]. The related include not only the governance of services and their choreographies, but also the governance of Service Level Agreements (SLA) and their Life Cycles. Figure 35 depicts the functionalities of the Governance & Management concern, as addressed in the CHOReOS project. The governance and management functionalities are related to the discovery concern presented in Section 4.4 as well as the analysis concern presented in Section 4.3.

In more detail, the functionalities embedded in the Governance & Management concern subdivide into:

- **Governance**, which further includes:
 - **Service Governance** that implements the **Service enforcement** (see Section 4.3) and the **Service Registry** (see Section 4.4) functionalities.
 - **SLA Governance** that encompasses the **SLA Management**, the **SLA enforcement** and the **SLA Registry** functionalities. The SLA Management functionality is responsible for providing abilities for **creating, updating, negotiating** and **removing** SLAs. The SLA Registry functionality enables the **discovery, publication** and **SLA lookup**. A service provider publishes services and SLA templates including the services behavior according to runtime parameters. The service consumers access the SLA registry and select the required SLA templates according to their needs. Once selected, the SLA templates are negotiated between service providers and consumers. The **SLA Enforcement** functionality allows verification and validation according to governance policies.
 - **Choreography Governance** that implements the **Choreography Registry** and the **Choreography Enforcement** functionalities. The Choreography Registry allows the **publication, discovery** and **lookup** of choreography models and samples. This may be useful at design time if we link the governance tool with the choreography modeling tool. This way, when designing choreographies, the designer may be inspired from existing models and samples. As Choreographies are considered as being compositions of services, many functionalities provided by Service Governance match the ones of choreography governance. Finally, the choreography enforcement functionality helps to apply governance policies to choreographies.
- **Policy management** that enables the **creation, update** and **removal** of policies. It also allows the **definition, update** and **removal** of governance rules for services, choreographies and SLAs.
- **Runtime Governance** that deals mostly with runtime concerns. It defines a **runtime management** functionality that is itself divided into the **monitoring** and the **test activation** functionalities. Test activation enables the verification of the respect of the SLAs constraints and policy rules at runtime, and relates to the Analysis concern presented in Section 4.3.

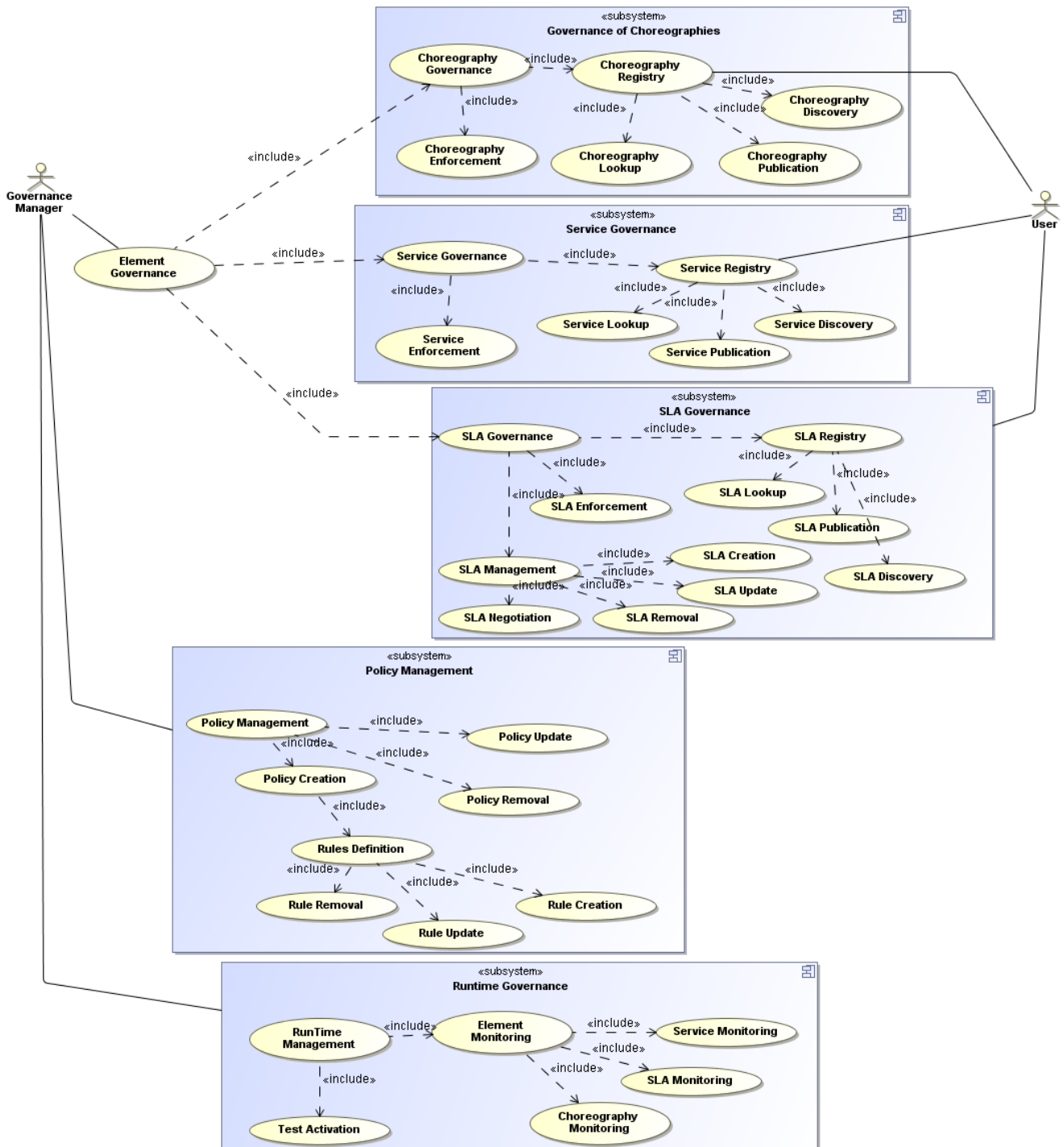


Figure 35: Functionalities of the Governance and Management Concern

4.8.1. Structure View

We extend the NEXOF reference diagram with classes expressing services, choreographies and SLAs, their life cycle and the governance concerns. Service Level agreement management has been addressed in the NEXOF Project as the management of the contracts between service providers and consumers. This functionality is also required in CHOReOS in order to allow the negotiations between services consumers and services providers. This will be extensively addressed in WP4 on Governance and V&V. The NEXOF Structure view is entirely revised to include functionalities needed by CHOReOS. Moreover, a modeling effort has been provided in order to generalize the structure view used in NEXOF and extend it to cover the Choreography concern.

The diagram in Figure 36 presents the structure view of the Governance and Management concern of the CHOReOS initial reference model. The view includes:

- **Element** where we consider the following elements: **choreographies**, **services** and **SLAs**. SLAs are composed of several **SLO** (Service Level Objectives) defining the **performance** and **usage metrics**. An element has a **life cycle** and can be described (**Description**). Finally, **policies** can be applied to elements.
- **LifeCycle** that is associated to each element. The lifecycle covers the elements from design to runtime and helps applying distinctly **governance policies** and **rules** according to each step of the lifecycle. Development, design, and runtime can be part of the life cycle structure. Further structures may be added if needed in order to cover the entire element lifecycle.
- **Policy** that specifies the behavior of an element. **Management**, **governance**, and **monitoring policies** then extend the Policy concept. Each Policy is composed by a set of rules that control the behavior of the elements from design- to run-time. Verification and validation rules are considered as belonging to the governance policy. These are related to the analysis concern.

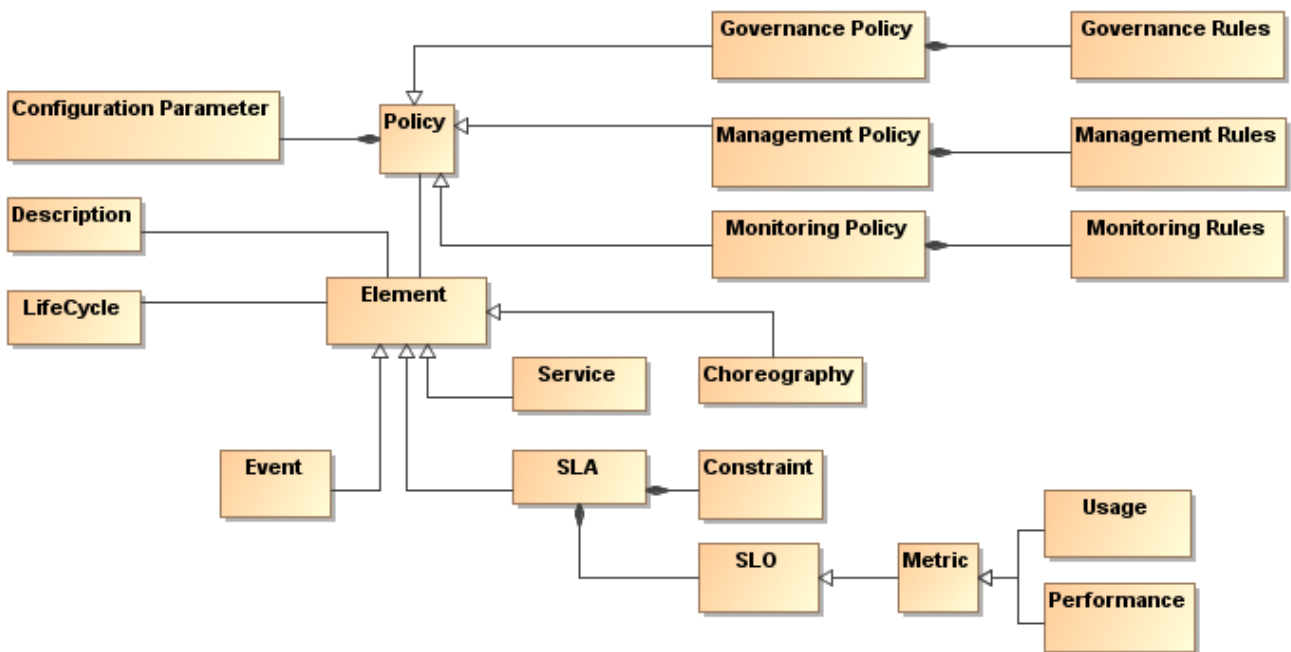


Figure 36: Structure View of the Governance and Management Concern

4.8.2. Data Flow

Following the structure view for the governance and management functionality, we present the data flow view for the main use cases.

Policy Management Functionality

The **policy management** functionality is presented in the data flow diagram in Figure 37. **Configuration parameters** are taken into consideration while the **policies** are applied. Policies can concern management, monitoring and governance. They are later applied to services, events, SLAs. Policies can be **created**, **added**, **updated** or **removed**.

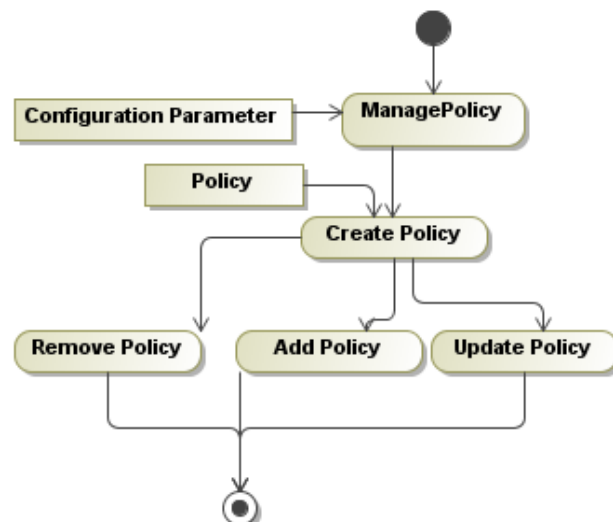


Figure 37: Functionality - Policy Management of the Governance and Management Concern

SLA Management functionality

The Service Level Agreement management activity enables the **creation**, **update**, **deletion**, and **negotiation** of **SLAs** (see Figure 38). SLAs templates are created by the service provider. The **SLO** and **metrics** are instantiated and their values set.

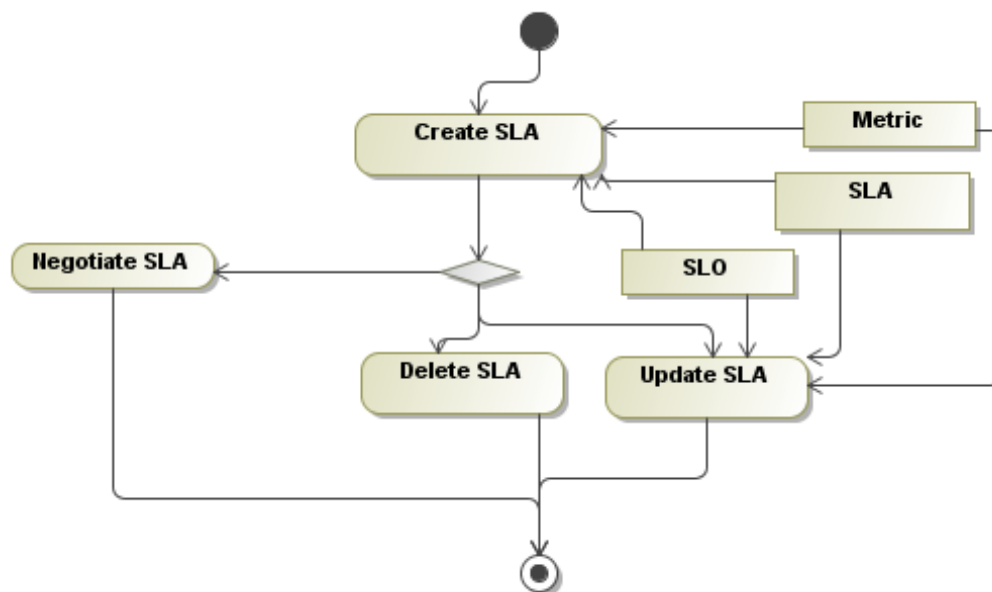


Figure 38: Functionality - SLA Management of the Governance and Management Concern

Monitoring Functionality

The **monitoring** functionality enables the observation of the services, choreographies and SLAs in the running environment (see Figure 39). **Notifications** can be triggered in order to inform the system about the gathered information. One can set control levels for each resource. The notifications are expressed into **Events**.

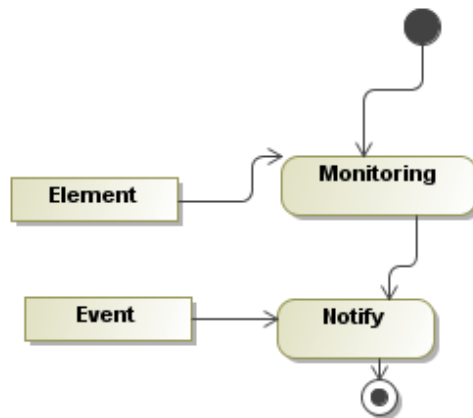


Figure 39: Functionality - Monitoring of the Governance and Management Concern

The presented data flow diagrams interact one with the other in order to provide the whole governance and V&V functionality.

4.9. Security Concern

The Security concern of NEXOF-RA covers important aspects such as Authentication, Authorization, Privacy and Integrity. In the initial CHOReOS Conceptual Model reported in this document, we adopt the Structure view and the Data Flow proposed in NEXOF-RA for the security concern, without adding specific CHOReOS extensions. Indeed, security, although a critical concern in the Future Internet, is not a focus of CHOReOS research. Further, in order to avoid duplication of efforts, we take a strategy of concertation with other projects in the FIA, whose focus is specifically in the security and trust areas. As a starting point, we will closely follow the conceptual framework under development within the NESSOS Network of Excellence, of which some CHOReOS partners are also members. In NESSOS, a specific workpackage is in fact dedicated to the creation of a common shared conceptual infrastructure for the Future Internet of Services.

5. Specifying and Realizing Concepts of the Initial Conceptual Model

The initial conceptual model presented in the previous section provides a preliminary integrated view of high-level concepts and functionalities in support of service choreographies. In this section, we present the baselines for various languages, models, technologies that can be used for choreography specification, execution, governance and V&V. In other words, this section serves introducing appropriate modeling languages and techniques for the concerns introduced by the conceptual model presented in the previous section.

As specific baselines, we consider:

- (i) An extension of *i** as graphical modeling technique [YMY94] to enable domain-expert high-level specification of choreographies in CHOReOS. The proposed extension will be proposed to support the functionality “Goal and Requirements Specification” of the Composition Concern;
- (ii) SoaML and USDL as modeling and description languages for service-oriented systems. Such languages can be used to describe aspects such as service behavior (e.g., “Service Behavioral Models” of the Composition Concern), as well as “Service Contracts” (e.g., “SLAs” of the Governance and Management Concern);
- (iii) BPMN2 as lower-level choreography specification language, supporting the functionality “Specification of Service Choreography” of the Composition Concern;
- (iv) Techniques to support the functionality “Enactment” of the Composition Concern, and
- (v) A layered approach to support the “Governance and Management Concern”.

All the above approaches have already been considered in [D1.1] from a more general point of view, whereas in the following, they are considered with the purpose of specifying and realizing concepts of the functionalities defined by the initial conceptual model.

5.1. Baseline for Domain Expert Specification of Requirements

To enable domain-expert specification of choreographies in CHOReOS, its services and associated qualities, we will extend the established *i** graphical modeling technique [YMY94] and its underlying GRL (Goal Representation Language) with syntax and semantics specific to service-oriented systems. Starting with end-user text-based requirements, these will be structured using the underlying GRL and semantics to model services, service roles and the dependencies between services in different choreographies. A meta-model for the *i** framework [FG09] is presented in Figure 40. We will use this as the baseline for a new meta-model reflecting our *i** extension for CHOReOS.

The central unit to be modeled in *i** is the strategic actor, with intentional aspects such as objectives, rationale and commitments. This permits the modeling of systems as networks of cooperating but interdependent heterogeneous actors, with intentional elements distinguished into four elements: goals, soft goals, tasks and resources. The network connections are represented by dependencies between actors, which indicate that one actor depends on another for the former to attain an intentional aspect [YMY94]. Our work will build upon this by introducing additional syntax and semantics to distinguish between consumer and service actors, and between services delivered by software, devices and humans. It will also introduce new semantics for modeling dependencies between actors that are specific to different choreography strategies, thereby describing different levels and forms of dependencies between actors fulfilled by services to undertake tasks and achieve goals.

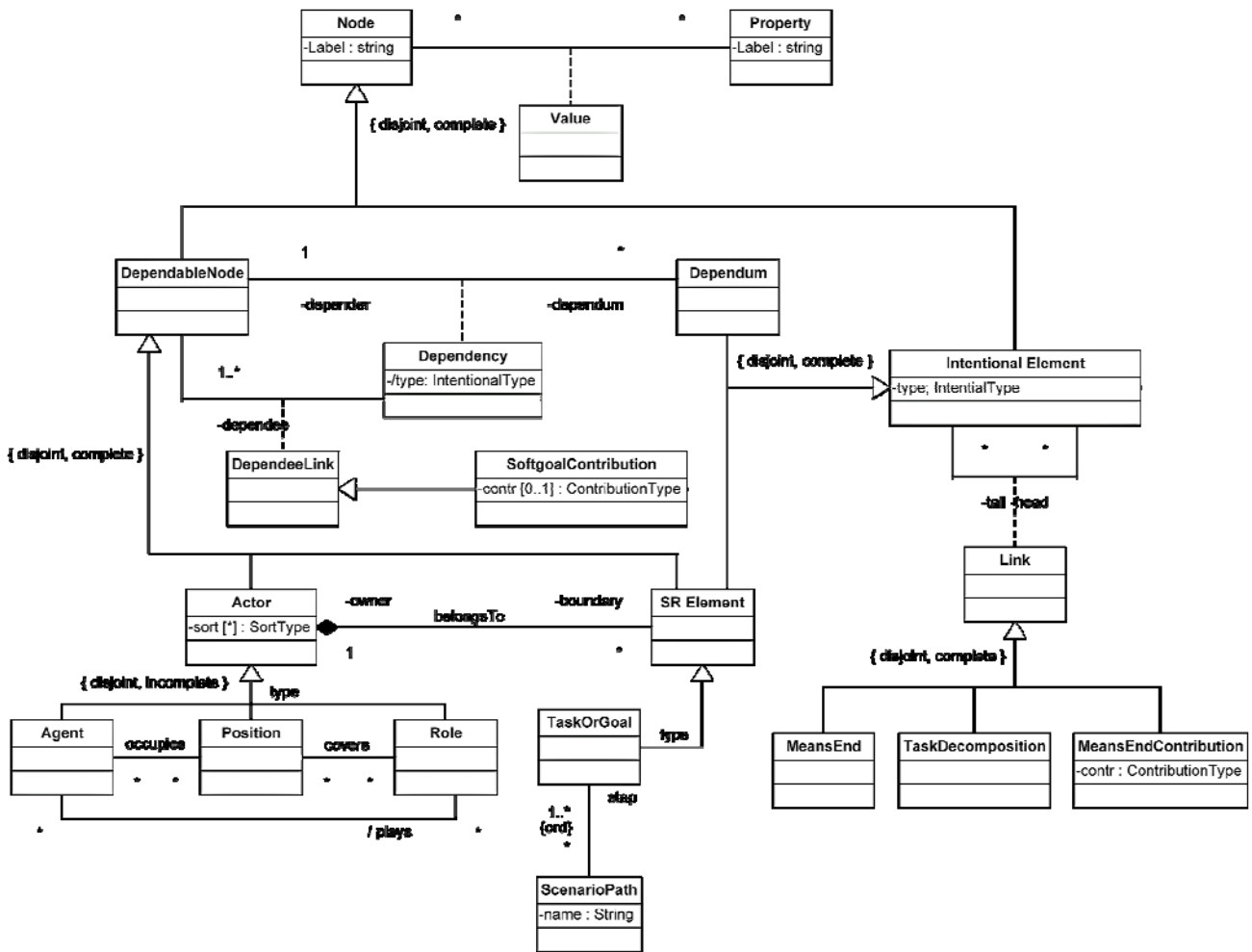


Figure 40: UML Class Diagram for the *i Framework (simplified for readability)**

Within this framework, we include the concept of a satisfaction argument to support reasoning about service and system qualities expressed for a service-oriented system. A satisfaction argument states that using the relevant properties of the application domain, when combined with the specification of the behavior of the system to be implemented, it is possible to show that the requirement will hold [HRH01]. We will tailor satisfaction arguments to address challenges specific to the choreography of service-oriented systems, and in particular we will use them to elaborate *i** means-end links between actor goals. Satisfaction arguments were first reported in [JAC95] and applied in the REVEAL requirements method [HRH01] to recognize the role of assumptions in specifications. This extension is not novel – previously we introduced satisfaction arguments to handle the role of domain knowledge in the specification of safety-related requirements in air traffic control [MLR07].

Figure 41 shows the conceptual model linking satisfaction arguments to *i**. An actor seeks to achieve or attain an end-element, which can be a goal or soft goal (for example a service quality). An actor also has the means to achieve or attain the end-element, usually a task (or functional requirement). Where the end element is a soft goal, the relationship is attributed with values that specify the modality and type of contribution (Some+, Some-, Help, Hurt, Make, Break, Unknown) reported in [YLL01]. Also, soft goals can represent tradeoffs, for example between service qualities. Each satisfaction argument is developed for one and only one end-element of a means-end link, and includes properties of the domain that must be true for the means to contribute to the end.

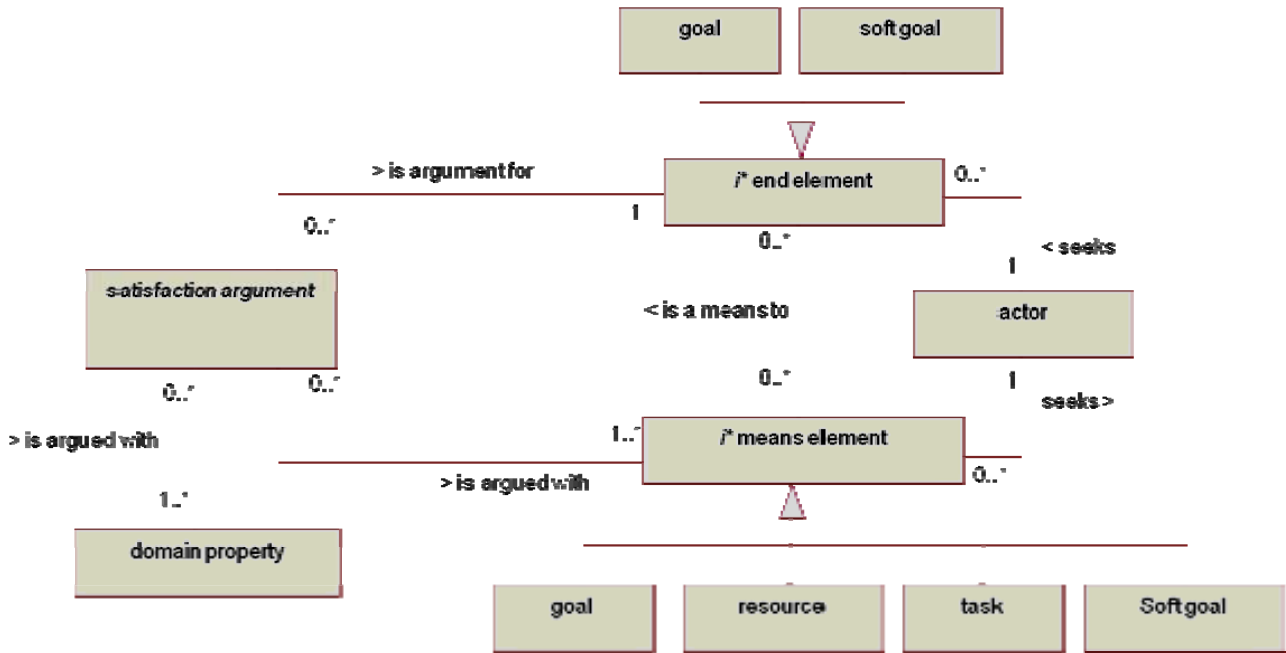


Figure 41: Conceptual Model that relates Concepts from *i and Satisfaction Argument**

Using this approach we will, for the first time, extend satisfaction arguments with information about choreography strategies and their indicative impacts on the relationships between the qualities of different services and, ultimately, the system. Central to the success of this technique is an understanding of how a specified choreography affects service qualities, hence we will develop simple libraries of patterns from empirical studies of choreographed systems to describe these affects and enable their reuse in satisfaction arguments. We will use these affects to propagate service qualities through a model of a service-oriented system based on tool-supported model walkthroughs. We will use a text-based representation of the underlying *i** model for the end-user to navigate, complete with the associated satisfaction arguments. The result will be a further extension to the new large-scale *i** modeling technique to introduce service choreography and explore its impact on a modeled service-oriented system.

Finally, we will link the extended *i** approach with a new quality ontology to express qualities that actors are seeking to achieve with consistent metrics and scales. Figure 42 shows an extract from the Service Measurement Index (SMI) *i** reference model, which specifies the definitions, measures, metrics and indicators of cloud service characteristics [CCC10]. Soft goals and contribution links represent the characteristics and relationships respectively, and task and resources represent measurements and metrics. We will build upon this research for CHOReOS, the result being a new version of the *i** modeling technique and its underlying semantics to model service-oriented systems and their qualities.

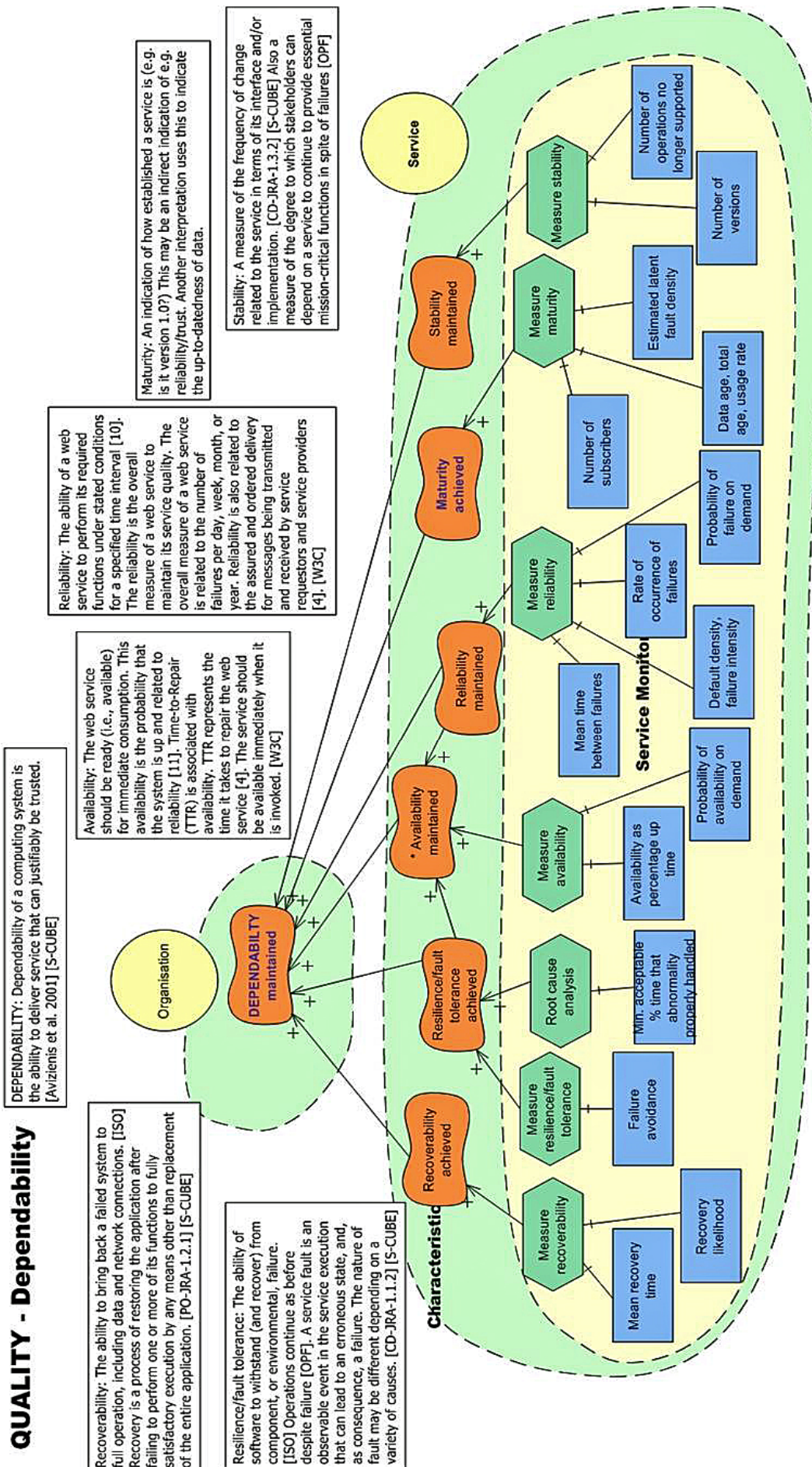


Figure 42: *i** example showing the Decomposition of the Dependability Characteristic for Cloud Services – representing Definitions, Measures, Metrics and Indicators

5.2. Baseline for Service Specifications

5.2.1. SoaML – Service-oriented architecture Modeling Language

The SoaML (Service-oriented architecture Modeling Language) is a new modelling standard that is under development in OMG (Object Management Group). SoaML aims to add SOA concepts on top of UML (Unified Modeling Language), which is considered as *de facto* standard in software modelling.

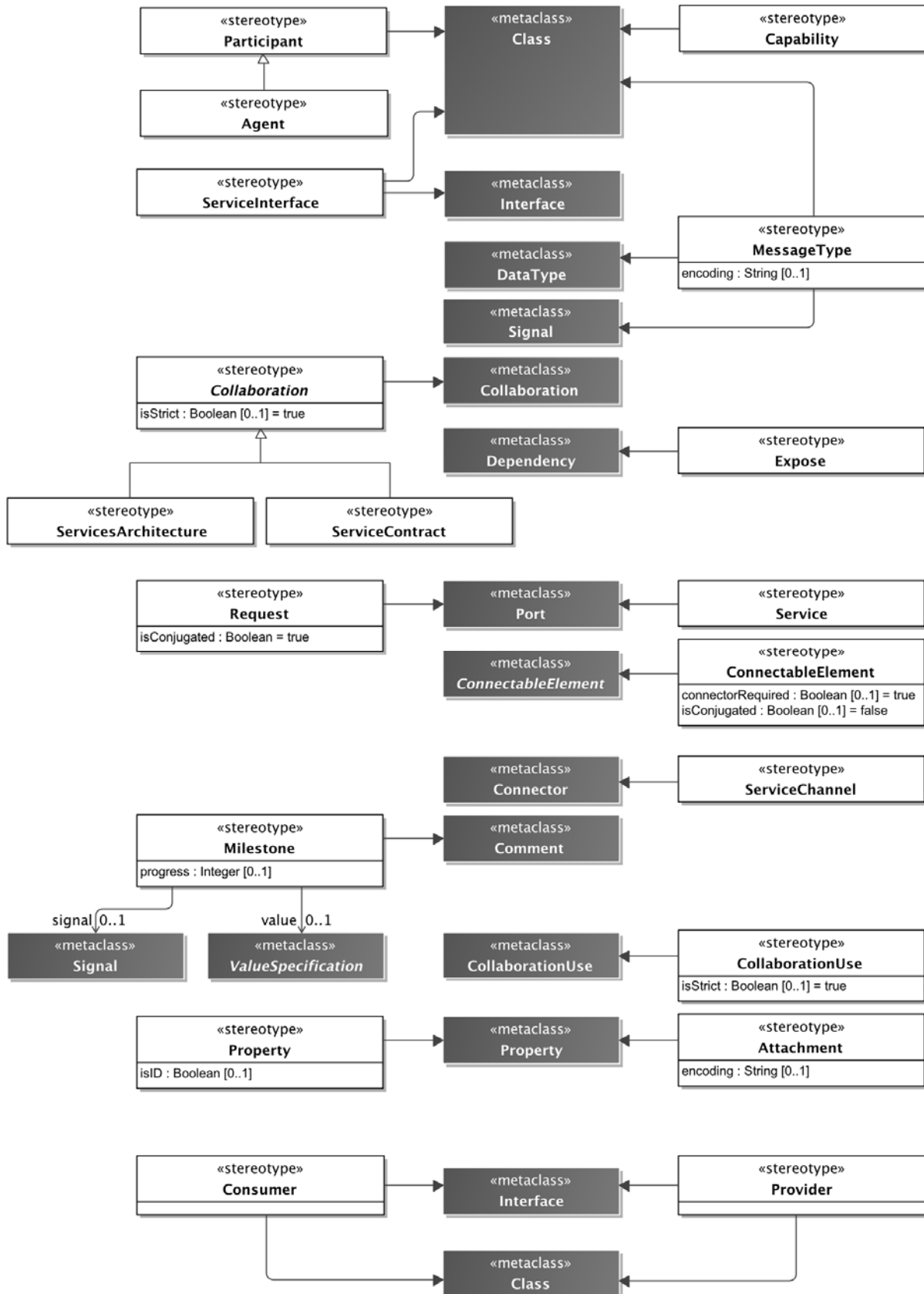


Figure 43: UML Profile for SoaML

SoaML is defined both as a metamodel, which extends the UML metamodel, and as a UML profile (Figure 43). SoaML is a rather light addition, which adds only few concepts to UML that mostly rely on the underlying UML concepts – only a few additional properties are added in several SoaML metaclasses/stereotypes.

Probably the most important SoaML aspect is that it provides *interoperability* with UML based models and other OMG modelling standards based on UML, e.g., SysML (System Modeling Language), which provides important Requirements concepts that are missing in UML. Thus, SoaML provides evolutionary instead of revolutionary approach to model-based SOA, which can be easily supported by modelling tool vendors and taken into use by modellers.

SoaML is a candidate for application in CHOReOS as a part of choreography design tooling. However, it does not provide all the necessary concepts, with the following most important limitations:

- SoaML does not provide a means for capturing requirements or integrating with external requirements models;
- SoaML does not provide a means and a clear approach for integration with business process models expressed in BPMN 2.0, which is considered as *de facto* standard in business process modelling.

CHOReOS aims to solve these limitations by combining SoaML with BPMN 2.0 and CHOReOS specific modelling concepts using UML profiles as model integration technology.

Also, it is worth mentioning that SoaML is still in *beta2* version, which was not finalized since the end of 2009 due to technical debates at OMG. One of the aims of the CHOReOS project is to contribute to the finalization and further evolution of the SoaML standard.

5.2.2. USDL - Unified Service Description Language

The Unified Service Description Language (USDL) [USDL10] is a platform-independent language for describing services. It is provided by SAP Research organization. USDL can also be a good candidate in CHOREOS for describing the services participating to a choreography.

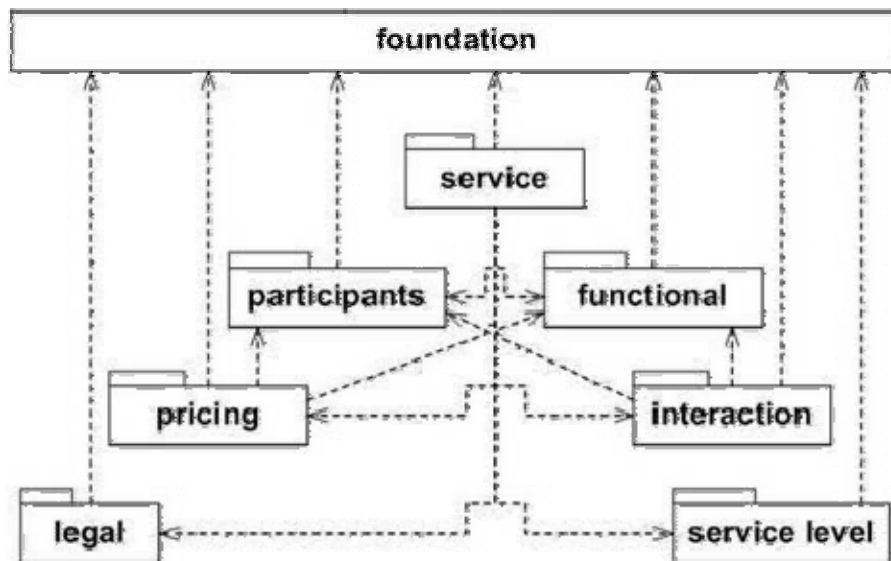


Figure 44: USDL Metamodel Modules

On a whole, USDL is made up of a set of modules, each addressing different aspects of the overall service description. USDL modules are inter-dependent, as they may reuse concepts from other modules. Figure 44 depicts the modules of the USDL platform:

- The topmost *Foundation* module covers common concepts that are used multiple times throughout other parts of USDL, concepts that cannot logically be assigned to any module, and concepts that are independent of the notion of service.
- The *Service* module represents the central concepts of USDL, which are the service and service bundle concepts; the module further includes some operational aspects such as the functional decomposition, and dependencies between services and other entities.

- The *Participants* module covers concepts that relate to the actors participating in the provisioning, delivery and consumption of services and service bundles.
- The *Functional* module allows capturing the business functions and capabilities of a service.
- The *Pricing* module covers the concepts that are needed to adequately describe price structures in the service industry. Given the increasing complexity and variety of this aspect of service provision, a modular representation of price charges is supported, together with the elements necessary to specify common segmentation strategies.
- The *Interaction* module covers concepts that describe how consumers should communicate with complex services with respect to individual interaction points.
- The *Legal* module provides description capabilities for usage rights. The module does not serve as a comprehensive contract template; in fact, such a contract is determined by several parameters that are derived from the other modules, in particular Pricing, Functional, and Service Level.
- The *Service Level* module covers concepts around Service Level Agreements (SLAs), i.e., the specification of agreed conditions for a service delivery, namely guaranteed states and guaranteed actions.

5.3. Baseline for Choreography Specifications

CHOReOS intends to provide tooling for choreography modeling, which will use BPMN (Business Process Model and Notation). BPMN is accepted as de facto standard in business process modeling, and its recent version 2.0 provides Choreography Diagrams (Figure 45). Since integration of business processes and software services is a hot topic, CHOReOS expects maximum impact by providing a choreography design solution, which will be based on extended version of BPMN 2.0.

In BPMN 2.0, choreography is a type of business process, but differs in purpose and behavior from a standard BPMN process. A BPMN standard process defines the flow of activities of a specific partner entity or organization. Contrariwise, *choreography formalizes the way business participants coordinate their interactions*. The focus is not on orchestrations of the activities performed within these participants, but rather on the *external exchange of information (messages) among them*. Thus, a choreography diagram is another way of modeling the “ping-pong game” among different participants without explicitly specifying the internal process flow of each participant.

BPMN 2.0 has evolved into a rather powerful and rich language with a well defined meta-model, which provides a rich set of process modeling concepts, but still only slightly touches other important aspects such as data structure modeling, which is extremely important for modeling service messages, or defining interfaces and quality contracts of software components providing services that are consumed in business processes. Unfortunately, BPMN was defined as a separate metamodel, which provides no interoperability with UML/SoaML models that can capture further details on service contracts and messages. CHOReOS intends to build UML profile for BPMN 2.0 choreography modeling capabilities, taking as a starting point the existing UML profile for BPMN 2.0 that is provided with MagicDraw modeling product line. However, this profile is not standardized at the moment. In order to achieve maximum adoption and impact, the CHOReOS project intends to contribute to the standardization of UML Profile for Business Processes at OMG. Such a standardized profile will not only enable business process choreography modeling in UML tools but also play a significant role in adopting BPMN 2.0, and choreography modeling in particular, as a part of UML based Enterprise Architecture (EA) frameworks, MODAF that is based on UPDM (Unified Profile for DoDAF and MODAF) [UPDM05].

CHOReOS will also go beyond the definition of UML profile for Business Processes, as it will be necessary to identify new concepts that are required to support user-centrism, decentralization, compositionality and incrementality, adaptation, and QoS awareness of choreographies. In particular, new modeling constructs will be conceived to deal with decentralization, compositionality and incrementality that are the fundamentals of ultra-large scale choreographies that CHOReOS wants to investigate. CHOReOS intends to promote some of these concepts to become a part of BPMN 2.x or later revisions.

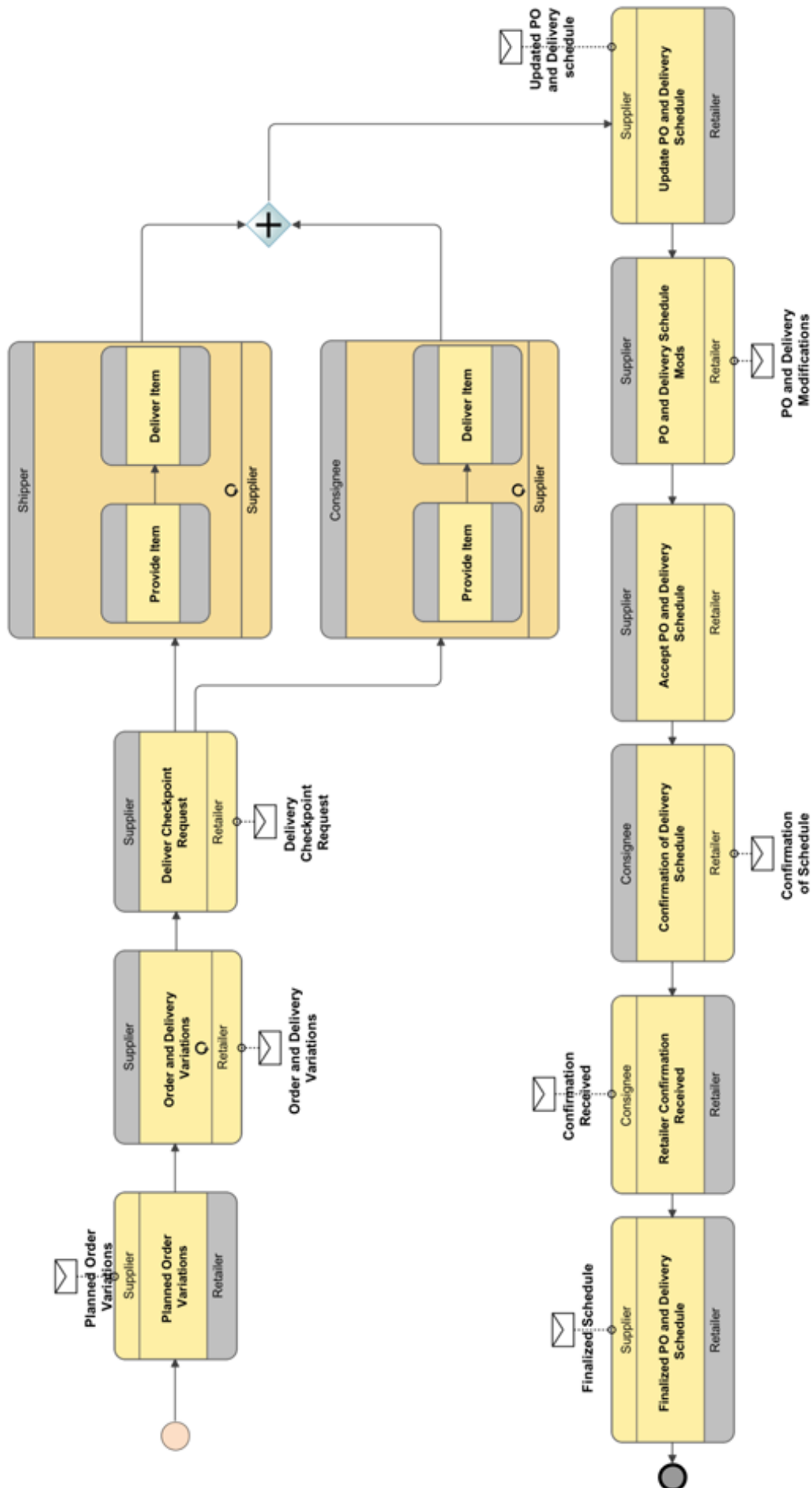


Figure 45: An example of BPMN 2.0 Choreography Diagram

5.4. Baseline for Choreography Execution

Service choreography gives a more distributed and scalable vision of service composition. Service choreography is not intended to be executed but to provide a global view of the concurrent execution of a number of interacting service processes. In ultra-large scale systems, due to the important number of processes that are running, it is essential to control services behavior during the execution. Service choreography can be seen as a governance mechanism that ensures the good function of the executing processes.

While service composition languages are numerous, there are only few languages for describing service choreography. Languages include the WS-CDL (Web Service Choreography Description Language) and the WSCI (Web Service Choreography Initiative) from W3C, which are implementation-focused. Let's dance [Zaha2006] is a graphical-oriented academic language. On the other hand, the BPMN2.0 standard integrates in its latest version choreography graphical annotations. Graphical annotations are interesting since they are platform-independent and they allow an easy representation of high-level choreography of services.

Approaches to service choreography address the multi-partners and distribution concerns that are required in ultra-large scale systems but which are not fulfilled by service orchestration. As systems are becoming more widespread and services more numerous, service composition needs to cope with large-scale systems and to tackle the raising complexity induced by heterogeneity and scalability. A centralized view is no more sufficient to answer new Future Internet system requirements. In CHOReOS, we deal with a widely distributed system view.

The authors in [Yildiz2007] address this issue by proposing a framework for process transformation technique. Their approach converts a centralized BPEL process into a set of nested processes able to be deployed on dynamically bound services. The authors consider a graph-like representation of the process and a transformation algorithm.

In [Violi2007], the authors consider a multi-agent system where BPEL activities are autonomously executed by agents and the mediation is supported via a coordination model. A semantic layer is also added to describe service. The overall engine architecture is composed of three tiers each responsible of a different aspect of the system: the interaction tier collects the incoming requests from the Web Service that is orchestrated by the BPEL; the correlation tier is responsible for interpreting messages and their correlation sets. Then, the messages are routed to the proper process instance. Finally, the workflow tier executes the activities of each process instance.

In [Chen2008], the authors develop a concurrent service orchestration engine called OnceBPEL2.0 based on Event-Driven Architecture. They target the improvement of performance under massive concurrency and design a scalable BPEL engine. Finite State machines are used to describe the semantics of the BPEL process. A mapping between BPEL activities and finite state machines is realized. Then, rules designing the state transitions within the process are represented using EDA. Agents are set in the orchestration engine to cope with time-consuming tasks. The authors evaluate their approach by realizing performance evaluations of average throughput and response time. The evaluation tests consider until two hundred virtual users.

In the CHOReOS project, challenging issues need to be addressed; these are related to the growing number of users, services and interactions. A high level modeling language that captures the user requirements is needed. The BPMN2.0 language previously presented is a good candidate for designing choreographies in a graphical way. However, BPMN2.0 specification is not executable and is not linked to executable orchestration processes.

For describing and executing the service choreographies, in the CHOReOS Project, we may consider both BPMN2.0 and the BPEL orchestration language. In Figure 46, we present a multilayer view of choreography and orchestration, where choreography specifies at a high level the global interactions of many participants without delving into the details. In a lower layer, distributed orchestrations are represented using BPEL. Execution is then handled via orchestration engines that must be monitored at a high level in order to ensure the good behavior of the choreography.

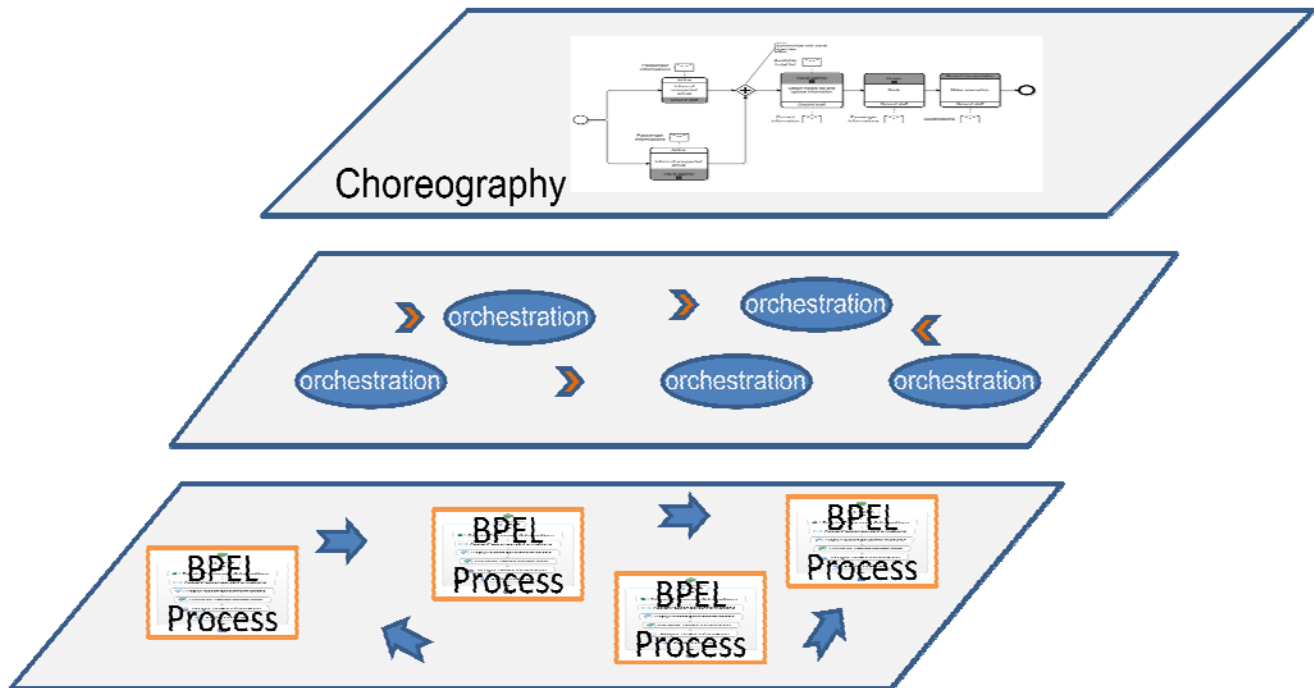


Figure 46: Multilayer View of Choreography and Orchestration

5.5. Baseline for Governance and V&V

A study conducted by Gartner lists insufficient validation in the “hit list” of the most common technological errors for planning SOA implementations [Gart07]. The same study recommends that at least 25% of the effort spent in a SOA project must be dedicated to testing. Although other European projects, such as SECSE and S-Cube, have analyzed SOA solutions, focusing on software testing and quality evaluation, a precise conceptual model has not been derived. Besides, the V&V concerns have not been so deeply investigated in the service engineering literature, although V&V activities can mainly be included in the Analysis concern suggested by the NEXOF-RA conceptual model.

In the CHOReOS approach for V&V, testing will play a very relevant and significant role. In that sense, the CHOReOS conceptual model incorporates design-time, development-time, and run-time testing as a fundamental activity of the CHOReOS development cycle.

Figure 47 depicts a global view of the governance baselines that can be adopted in CHOReOS. It is possible to subdivide governance into the following layers:

- **Services:** At the service level, governance deals with observing the services given their description. Web Service Description Language is a commonly used specification for describing web services. A WSDL-based service is described by a XML file which defines its location, interfaces, and operational data allowing its use. It helps services publication and discovery.
- **Service Level Agreement (SLA):** Contracts are made between a service provider and a service consumer. In these contracts, non-functional features such as performance and material resource usage can be mentioned as a warranty for the service consumer. The service level agreement is a standard that allows expressing this kind of contracts between services. The governance tool may consider also the publication and discovery of related SLOs to allow users to choose services considering non-functional concerns.
- **Choreography:** the governance of choreographies includes the definition of a choreography registry from which a choreography designer can take samples or models. Besides, governing choreography may include the definition of non functional concerns at design-time. As choreography is composed of a set of distributed services, service governance functionality may cover its main concerns.
- **Life Cycle:** Each element (service, choreography, SLA) has a life cycle that can be governed at each step from design time to runtime. Test activation including Verification and Validation is one of those steps; consequently we can consider a testing framework and methodology. At design-time, model, service, SLA, and choreography specifications will be checked for consistency and feasibility. At development-time, a Test-Driven Development (TDD) [Bec02]

methodology, supported by a generic framework for automated testing of choreographies, will be used to produce more reliable code with fewer bugs. At run-time, monitoring of choreographies will be carried out to validate the execution of the systems. Policies need also to be applied.

The governance and V&V tool must be interfaced with the service execution environment in order to monitor and control services, orchestrations and choreographies at runtime.

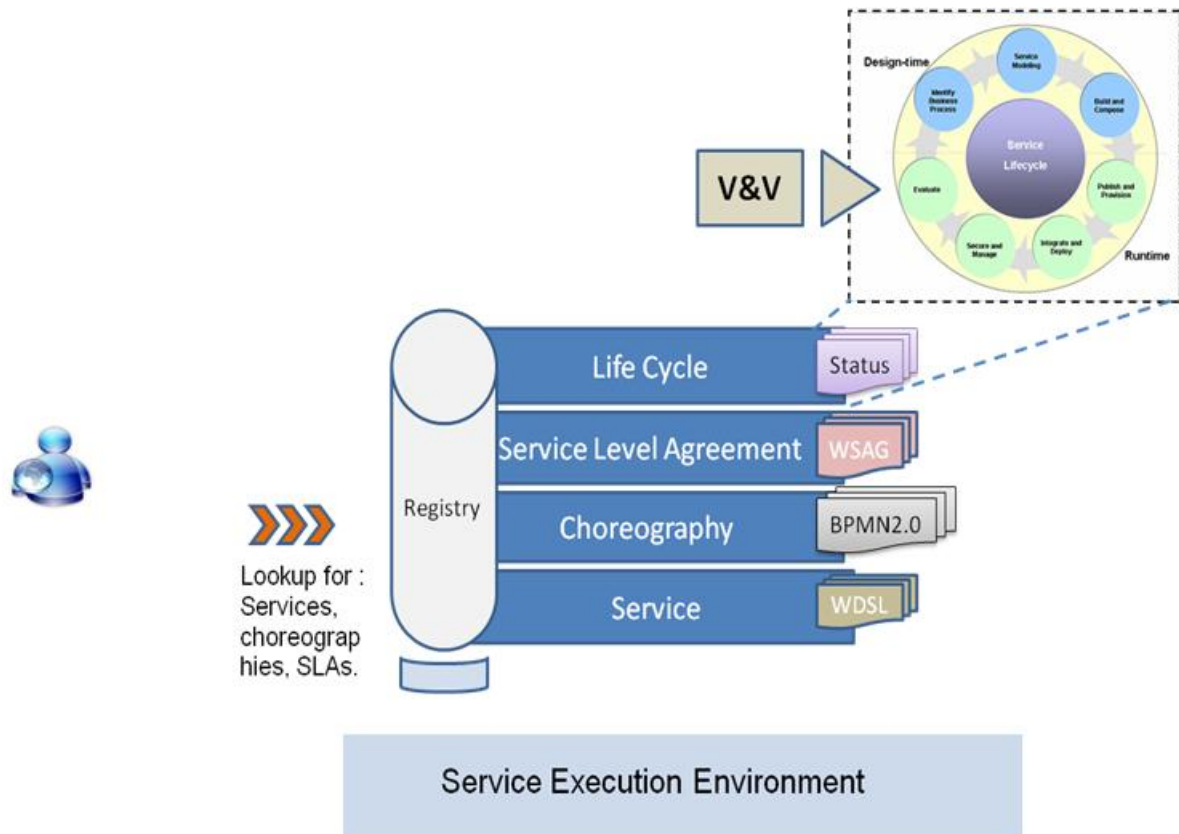


Figure 47: Baseline for Governance and V&V within CHOReOS

Finally, in the context of testing activities and the corresponding derivation of a conceptual model, UML Testing profile [UMLTest05] will be addressed. This is a general conceptual model that will help us to derive and elaborate a specific conceptual model within the context targeted by CHOReOS.

6. Conclusions

In this deliverable, we outlined the CHOReOS perspective on the choreography-based Future Internet, its requirements and its conceptualization. Based on the broad and mostly informal definition of the Future Internet available to us, we provided an overview of the many facets that must be considered in the realization of the Future Internet, and in particular the related Internet of Services and Things. By providing an overview independent of the more specific focus of CHOReOS, we were able to put the RTD work into the perspective of the overall Future Internet vision, and further elicit requirements for the choreography-based Future Internet where systems are realized as a composition of services discovered in ULS networking environments.

Considering NEXOF-RA as a praeludium providing general background concepts underpinning SOA infrastructures within the Future Internet, the initial Conceptual Model for CHOReOS has been defined according to the provided overview of the Future Internet and its requirements. Then, different languages, models, technologies have been considered as baselines to specify and realize concepts devised by the conceptual model, i.e., choreography and service specification, execution, governance, and V&V.

The initial CHOReOS Conceptual Model for the choreography-based Future Internet, introduced in this deliverable, will serve as input to the development of the RTD work packages WP2-3-4-5. The further definition and refinement of the conceptual model in WP1 will be undertaken in parallel with these work packages, hence benefiting from the lessons learnt from concrete instantiations as part of technological developments.

In more detail, based on the initial conceptual model introduced in this deliverable, our ongoing and follow-up work relates to:

- Studying the impact at large of the Future Internet's ULS dimensions and related challenges upon the traditional SOA paradigms so as to accordingly define in WP1 a reference architectural style for architecting CHOReOS-based systems. The CHOReOS style shall identify key aspects of the target systems, in terms of representative components (abstracting services in the Future Internet of Services and Things), connectors (abstracting interaction protocols) and coordination protocols (abstracting the behavior of choreographies);
- Devising in WP2 the CHOReOS development process and supporting toolset for choreography design;
- Devising in WP3 the CHOReOS middleware to provide the necessary runtime support for choreographies in the ULS Future Internet of Services and Things;
- Devising in WP4 the CHOReOS solutions supporting Governance and Verification & Validation of ULS choreographies;
- Defining in WP5 the infrastructure architecture of the CHOReOS IDRE, which integrates the results of WP2 to WP4, thereby supporting the design, development, enactment, and validation of choreography-based large scale service-oriented systems in the Future Internet.

The above will lead to refine the conceptual model, further resulting into strictly related artifacts that will characterize and support the choreography-based Future Internet.

7. References

[AL-MASRIM08]	E. Al-Masri and Q. H. Mahmoud. Investigating web services on the world wide web. In WWW, pages 795–804, 2008.
[AZI09a]	D. Athanasopoulos, A. Zarras and V. Issarny. ForeverSOA: Towards the Maintenance of Service Oriented Software. In Proceedings of the 3rd CSMR Workshop on Software Quality and Maintenance (SQM), 2009.
[BBBE07]	M. Beisiegel, H. Blohm, D. Booz, M. Edwards, O. Hurley, S. Ielceanu, A. Miller, A. Karmarkar, A. Malhotra, J. Marino, M. Nally, E. Newcomer, S. Patil, G. Pavlik, M. Raeppe, M. Rowley, K. Tam, S. Vorthmann, P. Walker, L. Waterman. SCA Service Component Architecture - Assembly Model Specification. Open Service Oriented Architecture www.osoa.org/download/attachments/35/SCA_Assembly_Model_V100.pdf?version=1 , 2007
[BDO06]	A. Barros, M. Dumas, and P. Oaks. Standards for Web Service Choreography and Orchestration: Status and Perspectives, presented at Business Process Management Workshops, Nancy, France, 2006
[Bec02]	K. Beck. Test-Driven Development: By Example. Addison-Wesley, 2002
[BHMN07]	D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris and D. Orchard. Web Services Architecture. Technical Report, W3C, 2004
[BIPT09]	A. Bertolino, P. Inverardi, P. Pelliccione and M. Tivoli, Automatic Synthesis of Behavior Protocols for Composable Web-Services, in: Proceedings of The 7th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE), Amsterdam, The Netherlands, pages 141-150, 2009
[Blackman10]	Colin Blackman, Ian Brown, Jonathan Cave, Simon Forge, Karmen Guevara, Lara Srivastava, Motohiro Tsuchiya and Rafael Popper, Towards a Future Internet Interrelation between Technological, Social and Economic Trends, 2010, Available at http://www.internetfutures.eu/wp-content/uploads/2010/11/TAFI-Final-Report.pdf .
[Carnot11]	Les Instituts Carnot, Smart Networked Objects and Internet of Things, 2011, Available at http://www.instituts-carnot.eu/files/AiCarnot-White_Paper-Smart_Networked_Objects_and_Internet_of_Things.pdf .
[CASAGRAS09]	CASAGRAS group. RFID and the Inclusive Model for the Internet of Things. http://www.rfidglobal.eu/userfiles/documents/FinalReport.pdf , 2009.
[CBDI07]	CBDI. CBDI-SAETM Meta Model for SOA Version 2.0. Technical Report, www.cbdiforum.com/public/meta_model_v2.php , 2007
[CCC10]	Cloud Commons Consortium, 2010, http://www.cloudcommons.com/
[CDDD05]	M. Colombo, E. Di Nitto, M. Di Penta, D. Distanto, M. Zuccalà. Speaking a Common Language: A Conceptual Model for Describing Service-Oriented Systems. In Proceedings of the 3rd International Conference on Service Oriented Computing (ICSOC 2005), Springer, 2005

[CISCO10]	Cisco: Cisco Visual Networking Index: Forecast and Methodology, 2009–2014, 2010, Available at http://www.cisco.com/en/US/solutions/collateral/-ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf .
[CISCO11]	Cisco, White paper Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010—2015, 2011, Available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/-white_paper_c11-520862.html .
[D1.1]	CHOReOS Project Team. CHOReOS State of the Art, Baseline, and Beyond. Public Project deliverable D1.1, 31 December 2010
[D5.1]	CHOReOS Project Team. Requirements for the CHOReOS IDRE. Public Project deliverable D5.1, 31 May 2011
[Daras09]	Daras P. et Al., Why Do We Need a Content-Centric Future Internet? Proposals Towards Content-Centric Internet Architectures, Information Society and Media Journal, 2009.
[DD04]	R. Dijkman and M. Dumas. Service-oriented Design: A Multi-viewpoint Approach. International Journal of Cooperative Information Systems, 13(4):337–368, 2004
[DSI08]	J.S. Da Silva, Eu approach towards the Future Internet. EURECOM message magazine (2008)
[EGA106]	Enterprise Grid Alliance. Reference Model and Use Cases - Part 1 of 2. EGA, 2006
[EGA206]	Enterprise Grid Alliance. Reference Model and Use Cases - Part 2 of 2. EGA, 2006
[eMobility07]	Frank-Uwe Andersen, Hendrik Berndt, Henrik Abramowicz, Rahim Tafazolli. Future Internet from Mobile and Wireless Requirements Perspective, 2007, Available at http://www.emobility.eu.org/WorkingGroups/PostIP/WhitePaper_v0-6.pdf .
[eMobility08]	Jose Marie Cabero, Tapio Frannti, Raffaele Giaffreda Jyrki Huusko, Luis Munoz, Rute Sofia, Rahim Tafazolli, Mika Ylianttila, Tanja Zseby, Future Internet in a Post-IP era and Experimental Test-beds, 2008, Available at http://www.emobility-ca.eu/deliverables/D3.2%20White Paper on Future Internet in a Post IP era FINAL.pdf .
[ETP09]	Future Internet, The Cross-ETP Vision Document, 2009, Available at http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf .
[Eurescom08]	Joao Schwarz da Silva. EU approach towards the Future Internet. EURECOM message magazine, pages 9-10, 2008.
[FANK05]	J. Fan and S. Kambhampati. A snapshot of public web services. SIGMOD Record, 34(1):24–32, 2005.
[FG09]	Franch, X., Grau, G. 'Towards a Catalogue of Patterns for defining Metrics over i* Models'. JISBD 2009: pp417-417, 2009
[FI_Vision]	Future Internet – The Cross-ETP Vision Document, v1.0, 2009. http://www.future-internet.eu/fileadmin/documents/reports/Cross-ETPs_FI_Vision_Document_v1_0.pdf
[FIA_FCN]	FIA – Future Content Network Scenarios, 2009. http://www.future-internet.eu/fileadmin/documents/prague_documents/FIA-FCN Internet Scenarios_20090507.pdf

[FIA09]	G. Tselenis et al. (Eds). Towards the Future Internet. IOS Press. 2009. http://www.future-internet.eu/publications/fia-book-2009.html .
[FIAFCN07]	FIA Future Content Networks (FCN), Scenarios, May 2009, http://www.future-internet.eu/fileadmin/documents/prague_documents/FIA-FCN_Internet_Scenarios_20090507.pdf
[FIAMANA09]	FIA Management and Service-aware Architectures (MANA), Scenarios for Future Internet, May 2009, http://www.futureinternet.eu/fileadmin/documents/prague_documents/MANA_Scenarios-Final.pdf
[FIArch11]	Ec FIArch Group, Fundamental Limitations of Current Internet and the Path to Future Internet, 2011, Available at http://ec.europa.eu/information_society/activities/foi/library/docs/fiarch-current-internet-limitations-march2011.pdf .
[FIRE09]	FIRE Use Scenarios (FIRE), May 2009, http://www.ict-fireworks.eu/fileadmin/events/FIA-Prague/1_Avessta.pdf
[FISE09]	FIA Future Internet Socio-Economics (FISE), FISE Scenarios and Effects, February 2009, http://www.smoothit.org/wiki/pmwiki.php/FISE/Scenarios
[FISO09]	FIA Future Internet Service Offer (FISO), Scenarios, February 2009, http://services.future-internet.eu/index.php/FISO_Scenarios
[FMI10]	Future Media Internet task force, Future Media Internet Research Challenges and the Road Ahead, 2010, Available at http://www.future-internet.eu/uploads/media/FMI-TF-White_paper_042010_01.pdf
[Gart07]	Gartner. 2007 Press Release: Bad Technical Implementations and Lack of Governance Increase Risks of Failure in SOA Projects. http://www.gartner.com/it/page.jsp?id=508397
[GRAPI11]	N. Georgantas, M. A. Rahaman, H. Ameziani, A. Pathak and V. Issarny. A Coordination Middleware for Orchestrating Heterogeneous Distributed Systems. In Proc. 6th International Conference on Grid and Pervasive Computing (GPC 2011). LNCS 6646. Oulu, Finland, May 2011.
[Hirabaru08]	AKARI Architecture Design Project: New Generation Network Architecture: AKARI Conceptual Design (ver1.1), 2008, Available at http://nag.nict.go.jp/topics/AKARI_fulltext_e_translated_version_1_1.pdf .
[HMY06]	G. Huang, H. Mei, and F. Yang. Runtime Recovery and Manipulation of Software Architecture of Component-based Systems. International Journal of Automated Software Engineering, 13(2), 2006.
[HRH01]	J. Hammond, R. Rawlings and A. Hall. Will It Work?, Proceedings 5th IEEE International Symposium on Requirements Engineering, IEEE Computer Society, 102-109, 2001
[HS10]	Pieter Hens, Monique Snoeck, Manu de Backer, and Geert Poels. Decentralized Event-based Orchestration in edBPM 2010 Proceedings of the 4th international workshop on Event Driven Business Process Management, pages 1-12, Hoboken, NJ, USA, 2010, BPM 2010.
[IOT08]	Bassi, A., Europe, H., Horn: Internet of Things in 2020: Roadmap for the Future, Internet of Things workshop report, 2008, Available at http://www.iot-visithethefuture.eu/fileadmin/documents/researchforeurope/270808_IoT_in_2020_Workshop_Report_V1-1.pdf .
[JAC95]	M. Jackson. "Software Requirements and Specifications". Addison-Wesley, 1995

[Jain06]	Raj Jain. Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation. MILCOM, pages 1-9, 2006.
[KIMR04]	S. M. Kim and M.-C. Rosu. A survey of public web services. In WWW (Alternate Track Papers & Posters), pages 312–313, 2004.
[Kortuem10]	Gerd Kortuem, Fahim kawsar, Vasughi Sundramoorthy, Daniel Fitton. Smart objects as building blocks of the internet. IEEE Internet Computing, pages 44, 2010.
[Kumar10]	Kumar, K. and Lu, Y.H., Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?, Computer IEEE, pages 51-56, 2010.
[Lassila07]	Lassila, O. and Hendler, J., Embracing “Web 3.0”, IEEE Internet Computing, pages 90-93, 2007.
[LILZLXS07]	Y. Li, Y. Liu, L.-J. Zhang, G. Li, B. Xie, and J. Sun. An exploratory study of web services on the internet. In ICWS, pages 380–387, 2007.
[Marks2008]	Eric A. Marks, Service Oriented Architecture Governance for the Services Driven Enterprise, John Wiley & Sons, Inc. Editions, 2008
[MH06]	H. Muccini and R. Hierons (Eds). ROSATEA 2006: The Role Of Software Architecture in Testing and Analysis. ACM Digital Library, 2006.
[MLR07]	N.A.M. Maiden, J. Lockerbie, D. Randall, S. Jones and D. Bush. Using Satisfaction Arguments to Enhance i* Modelling of an Air Traffic Management System. Proceedings 15th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 49-52, 2007
[MLR07]	N.A.M. Maiden, J. Lockerbie, D. Randall, S. Jones and D. Bush. Using Satisfaction Arguments to Enhance i* Modelling of an Air Traffic Management System. Proceedings 15th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 49-52, 2007
[NEXOFAFP10]	NEXOF-RA project (http://www.nexof-ra.eu/). Deliverable D7.2c: Definition of an architectural framework & principles, 2010. http://www.nexof-ra.eu/sites/default/files/D7.2c%20Definition%20of%20an%20architectural%20framework%20and%20principles%200.9.pdf
[NEXOFPI10]	S. Agassi, P. Bisson, J. Manuel C. Fonseca, R. Jimenez-Peris, A. De Nigro, D. Desideri, P. Corte. Pattern Internet of Service, http://www.nexof-ra.eu/sites/default/files/Internet%20of%20Service%200.6.pdf , 2010.
[NEXOFRA_D63]	NEXOF-RA Deliverable D6.3 NEXOF Reference Model, 2010. http://www.nexof-ra.eu/sites/default/files/D6.3_v1.0.pdf
[NEXOFRA_D73]	NEXOF-RA Deliverable D7.3 Conceptual architectural view, 2008. http://www.nexof-ra.eu/sites/default/files/D7.3_Conceptual_Architectural_View_v1.0_0.doc
[NEXOFRA_D74]	NEXOF-RA Deliverable D7.4 RA Specification Baseline v0.45, 2008. https://files.morfeo-project.org/mymobileweb/public/NEXOF-RA/D7.4/D7.4%20-%20RA%20Specification%20Baselinev0.45-THA.doc
[NEXOFRAS10]	NEXOF-RA project (http://www.nexof-ra.eu/). Deliverable D7.5b: RA Specification V1.0 (Pattern Compass), 2010. http://www.nexof-ra.eu/sites/default/files/D7.5b%20RA%20Specification%20V0.4a%20(Pattern%20Compass).pdf
[NEXOFRM10]	NEXOF-RA project (http://www.nexof-ra.eu/). Deliverable D6.3: The NEXOF-RA Reference Model V3.0. 2010 http://www.nexof-ra.eu/sites/default/files/D6.3_v1.0.pdf

[OAS06]	C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz. OASIS Reference Model for Service Oriented Architecture 1.0. Committee Specification 1, http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf , 2006
[Pap09]	D. Papadimitriou, Future Internet{The Cross-ETP Vision Document. European Technology Platform, Alcatel Lucent 8 (2009)
[Pereira08]	Jorge Pereira. From autonomous to cooperative distributed monitoring and control: Towards the Internet of smart things. ERCIM Workshop on eMobility, 2008.
[Plastic08]	PLASTIC consortium. "A B3G Service Platform: The IST PLASTIC Project". PLASTIC White Paper, http://plastic.paris-rocquencourt.inria.fr/plasticwhitepaper.pdf , 2008.
[PlasticWS]	PLASTIC consortium. The PLATIC web site: http://plastic.paris-rocquencourt.inria.fr/
[RWI09]	FIA Real World Internet (RWI), Scenarios, March 2009, Available at http://rwi.future-internet.eu/index.php/RWI_Scenarios
[Schonwalder09]	Schonwalder, J. and Fouquet, M. and Rodosek, G. and Hochstatter, I., Future Internet= Content+ Services+ Management, Communications Magazine, IEEE, pages 27-33, 2009
[SeCSE]	SeCSE, Project web site: http://www.secse-project.eu/
[SF09]	Service Finder: Realizing Web Service Discovery at Web Scale, February 2009, http://www.service-finder.eu/attachments/D8.4.pdf .
[SLC10]	Vanessa Strickera, Kim Lauenrotha, Piero Corteb, Frédéric Gittlerc, Stefano De Panfilisb, and Klaus Pohl, Creating a Reference Architecture for Service-Based Systems A Pattern-Based Approach. DOI:10.3233/978-1-60750-539-6-149. Towards the Future Internet, G. Tselentis et al. (Eds.) IOS Press, 2010
[SOA4All_D111]	SOA4All Deliverable D1.1.1 Design Principles for a Service Web v1, 2008. http://www.soa4all.eu/file-upload.html?func=startdown&id=25
[SoaML]	http://www.omg.org/spec/SoaML/
[Stuckmann09]	Peter Stuckmann, Rainer Zimmermann. European research on Future Internet design. IEEE Wireless Communication, pages 14-22, 2009.
[SWHA05]	L. Schubert, M. Wilson, J. Haller, A. Arenas, A. Svirskas, P. Giambiagi, J. Doser, E. Lupo. The TrustCoM Conceptual Models. 2005
[TFI10]	Towards the Future Internet - Emerging Trends from European Research. Editors: Georgios Tselentis, Alex Galis, Anastasius Gavras, Srdjan Krco, Volkmar Lotz, Elena Simperl, Burkhard Stiller, Theodore Zahariadis. DOI:10.3233/978-1-60750-539-6-149. IOS Press, 2010
[Theodore09]	Zahariadis, T.B. and PAU, G. and CELETTO, L. and DARAS, P., Seamless Content Delivery in the Future Mobile Internet, IEEE Wireless Communications, 2009.
[Thompson05]	Graig W. Thomson, Smart devices and soft controllers. IEEE Internet Computing, pages 82-85, 2005.

[TII08]	M.Tivoli and P.Inverardi. Failure-free coordinators synthesis for component-based architectures, Science of Computer Programming, 71(3), pp. 181-212, May 2008. DOI REF: http://dx.doi.org/10.1016/j.scico.2008.03.001
[Toma09]	Ioan Toma, Elena Simperl, Agata Filipowska , Graham Hench , John Domingue. Semantics-Driven Interoperability on the Future Internet. International Conference on Semantic Computing, 2009.
[TSF09]	I. Toma, E. Simperl, A. Filipowska, G. Hench, Domingue, J.: Semantics-Driven Interoperability on the Future Internet. In: IEEE International Conference on Semantic Computing, (ICSC) (2009)
[UDDI03]	OASIS group. Universal Description, Discovery and Integration (UDDI), version 3.0.1. Technical report, OASIS group, 2003. http://uddi.xml.org/
[ULS06]	Ultra-Large-Scale Systems. The Software Challenge of the Future. Software Engineering Institute, Carnegie Mellon. Pittsburgh, PA., 2006.
[UMLTest05]	OMG, UML Testing Profile Version 1.0 formal/05-07-07
[UPDM05]	OMG, Unified Profile For The Department Of Defense Architecture Framework (DoDAF) And The Ministry Of Defence Architecture Framework (MODAF), http://www.omg.org/spec/UPDM/
[USDL10]	http://www.internet-of-services.com/index.php?id=288&L=0 Unified Service Description Language available at http://www.internet-of-services.com/fileadmin/IOS/user_upload/pdf/USDL-3.0-module-core-M2-20091228.pdf Décembre 2009 SAP research
[Viroli2007]	Viroli, Mirko and Denti, Enrico and Ricci, Alessandro, Engineering a BPEL orchestration engine as a multi-agent system, In proceedings of Sci. Comput. Program Journal, May, 2007, p226—245, Elsevier North-Holland, Inc.
[XML09]	W3C: XML Schema Definition Language, version 1.1, 2009. http://www.w3.org/XML/Schema .
[Yang, 2007]	Yang, H.; Zhao, X.; Cai, C. & Qiu, Z. Exploring the Connection of Choreography and Orchestration with Exception Handling and Finalization/Compensation FORTE, 2007, 81-96
[Yildiz2007]	Yildiz, Ustun and Godart, Claude Towards decentralized service orchestrations Proceedings of the 2007 ACM symposium on Applied computing 2007 Seoul, Korea, p 1662—1666
[YLG 07]	Ustun Yildiz INRIA - LORIA, Gabriel Lippmann, Claude Godart Towards decentralized service orchestrations Symposium on Applied Computing Proceedings of the 2007 ACM symposium on Applied computing Seoul, Korea SESSION: Web technologies Pages: 1662 - 1666 Year of Publication: 2007 ISBN:1-59593-480-4
[Zaha2006]	J. M. Zaha and A. Barros and M. Dumas and A. terHofstede Let's Dance: A Language for Service Behavior Modeling Proceedings of the 14th International Conference on Cooperative Information Systems, (CoopIS) 2006
[Zhang10]	Zhang, S. and Zhang, S. and Chen, X. and Huo, X., Cloud Computing Research and Development Trend, 2010 Second International Conference on Future Networks, IEEE, pages 93-97, 2010.